Universidad Carlos III de Madrid

Escuela Politécnica Superior

Grado en Ingeniería Telemática

PRUEBAS DE MOVILIDAD BASADAS EN PMIP EN UN EMULADOR DE REDES

TRABAJO FIN DE GRADO

Autor: Enrique Esteso Saiz

Tutor: Dr. Ignacio Soto Campos

Leganés. Septiembre 2016

Trabajo Fin de Grado

PRUEBAS DE MOVILIDAD
BASADAS EN PMIP, EN UN EMULADOR DE REDES

Autor
Enrique Esteso Saiz

Tutor
Dr. Ignacio Soto Campos

Realizado el acto de defensa y lectura del Trabajo Fin de Grado el día de octubre de 2016 en Leganés; Escuela Politécnica Superior de la Universidad Carlos III de Madrid.

EL TRIBUNAL:

Presidente:

Secretario:

Vocal:

Acuerdan otorgarle la calificación de:

CALIFICACIÓN:

Leganés a de octubre de 2016

A mis padres y mi hermano

Agradecimientos

A todos mis profesores que desde la Educación Primaria han ayudado a mi formación, en especial a Leo y Jesús, mis primeros profesores de informática, con los que aprendí las posibilidades de este campo.

A mi tutor Ignacio, por su paciencia e interés en ayudarme a realizar este proyecto.

A mi padre al que siempre vi dedicado al mundo de los ordenadores y despertó en mí el interés por ellos.

A mi madre y mi hermano por su ayuda y ánimos en los momentos difíciles de mi carrera.

A todos aquellos que han estado en mi vida durante mis años universitarios y con los que he compartido los buenos y malos ratos que en ellos ha habido.

Por último, a la empresa Zennio, por su buena acogida y en especial a mi tutor de prácticas Juanjo, por sus enseñanzas tanto tecnológicas como humanas en el mundo profesional.

“Hay una fuerza motriz más poderosa que el vapor, la electricidad y la energía atómica: la voluntad”

Albert Einstein (1879- 1955)

ÍNDICE

RESUMEN....9
ABSTRACT....10
I. INTRODUCIÓN....12
I.1 - Generalidades....12
I.2 – Motivación....12
I.3 – Objetivos....13
I.4 – Estructura de la memoria....13
I. INTRODUCTION....15
I.1 - Generalities....15
I.2 – Motivation....15
I.3 – Objectives....16
I.4 – Memory Structure....16
II. ENTORNO SOCIOECONOMICO....19
III. MARCO REGULADOR....22
IV. ESTADO DEL ARTE....24
IV.1 Opciones estudiadas....24
IV.1.1 Emulador vs simulador de redes....24
IV.1.2 Emulador o distribución en máquinas físicas....24
IV.2 Análisis de herramientas....25
IV. 2.1 Emulador CORE....25
IV.2.2 Protocolo PMIPv6....29
V. DESCRIPCIÓN DEL TRABAJO REALIZADO....36
V.1 Preparación del entorno....36
V.2 Puesta en funcionamiento....36
V.3 Pruebas finales....52
V.4 Conclusiones....60
VI. CONCLUSIONES....62
VI. CONCLUSIONS....63
VII. LÍNEAS FUTURAS DE TRABAJO....65

VIII. PLANIFIACACIÓN DE TAREAS Y PRESUPUESTO 67
VIII.1 Equipamiento 67
VIII. 2 Tiempo de trabajo 67
VIII. 3 Costes indirectos 68
IX. BIBLIOGRAFÍA 71
ANEXO I 73
A) Configuración del Kernel de Linux. 73
B) Instalación y ejecución de PMIPv6 75
ANEXO II. PALABRAS CLAVE 78
ANEXO III. SUMMARY IN ENGLISH 81

RESUMEN

El objetivo principal de este trabajo es conseguir tener un entorno virtual en un emulador de redes denominado CORE, de tal forma que el protocolo PMIPv6 se esté ejecutando y funcione correctamente según lo estipulado en la RFC 5213. De esta forma conseguimos crear un entorno virtual que nos permite hacer las pruebas necesarias de movilidad en redes IP.

En la actualidad ha aumentado exponencialmente el uso de dispositivos móviles necesitados de conexión continua a internet. Por tanto los usuarios de estos dispositivos demandan conexiones móviles sin interrupciones en la transferencia de datos, ya que además el uso de algunas aplicaciones se hace imposible con conexiones que no sean constantes.

Para dar solución a este problema se creó PMIPv6. Un protocolo que gestiona la movilidad de los dispositivos a nivel IP y cuya gestión se realiza desde la red. Lo cual implica una gran ventaja sobre otros protocolos o tecnologías ya que no necesita de software adicional en los dispositivos finales.

Los resultados de este trabajo son satisfactorios en cuanto que las pruebas de funcionalidad y conexión en distintos tipos de escenarios han sido positivas. Aunque hay pequeñas desviaciones entre el estándar de PMIPv6 y el código usado, estas no afectan ni a la funcionalidad ni a la conexión entre los dispositivos.

ABSTRACT

The main objective of this Project is to get a virtual enviroment with in an nets emulator called CORE, then so PMIPv6 protocol is being executed and working properly accordingly with established in the RFC 5213. With this we are able to crate a virtual enviroment that allow us to make all the test of mobility in net IP we need.

Currently use of mobile devices with constant connection has increased exponential. Users demand internet conexions without cuts in transfering data, also because some of the applications can´t be used if connection is discontinued at any point of the process.

To solve this problema a PMIPv6 was created. A protocol that manages the mobility of the device to the IP level and which managment is made from the net. That means a huge advantage over other protocols or tecnologies because it does not need an aditional software in the final device.

Results of this work are satisfactories as to functionality and connection tests done in differents scenaries have been positive. Although there are Little deviations between the standard of PMIPv6 and the code used, these deviations do not affect the functionality or the connection among devices.

CAPÍTULO I

INTRODUCCIÓN

I. INTRODUCIÓN

I.1 - Generalidades

Este trabajo basa su interés en la necesidad de movilidad sin pérdidas de conexión a internet de los usuarios de dispositivos móviles. Se busca un entorno de pruebas en el que la ejecución del protocolo PMIPv6 sea sencillo de analizar y además se pueda ejecutar en escenarios diferentes con la ventaja de usar una única maquina física.

En esta introducción explicaremos:

- Motivación
- Objetivos
- Estructura de la memoria
-
  - Estado del arte
  - Marco regulador
  - Entorno socioeconómico
  - Descripción del trabajo realizado
  - Conclusiones
  - Líneas futuras de trabajo
  - Planificación de tareas y presupuesto
  - Bibliografía
  - Anexo1
  - Anexo2
  - Anexo3

I.2 – Motivación

Uno de los mayores retos de la tecnología de hoy es conseguir que todos los dispositivos móviles estén constantemente conectados a internet, evitando espacios de tiempo en los que el dispositivo quede sin conexión o consiguiendo que estos espacios sean lo más pequeños posibles. Esto se debe a la necesidad de las personas a usar los dispositivos continuamente y a que estas cada vez se mueven más rápido, desde coches hasta aviones. Este problema es más sencillo de solucionar cuando los dispositivos se mueven a baja velocidad, pero a medida que esta aumenta, los cambios se dan con mayor frecuencia y el problema se vuelve mayor ya que tenemos muchos más cambios de conexión con los puntos de acceso.

Por otro lado tendríamos el problema de la cantidad de dispositivos conectados. Si bien un dispositivo en una red local puede tener una dirección privada, cada vez son más los

dispositivos conectados necesitados de direcciones públicas las cuales deben ser únicas. La aparición de estos últimos ha aumentado exponencialmente en los últimos años y sigue aumentando, por lo que es necesario un sistema que gestione este aumento con eficiencia.

Ambos problemas son resueltos por PMIPv6, ya que por un lado es un protocolo diseñado específicamente para resolver el problema de la movilidad y además está basado en IPv6 que usa direcciones de 128 bits.
Basándonos en la necesidad de movilidad de los dispositivos en redes IP en la actualidad, parece interesante poder facilitar las pruebas con un protocolo como PMIPv6. La ventaja de este protocolo, frente a otros propuestos para el mismo propósito, es que no es necesario instalar software adicional en los dispositivos móviles, además de estar estandarizado para gestionar la movilidad en redes de tercera generación y siguientes, en concreto en redes 3GPP [1].

Otra motivación, es la posible utilidad de este proyecto en la docencia, dado que al ejecutar el protocolo en una única maquina con un emulador, los recursos utilizados son menores y la facilidad de uso, análisis y demostración de su ejecución es mucho más sencilla.

I.3 – Objetivos

El objetivo de este proyecto es conseguir que el protocolo PMIPv6 funcione correctamente en un emulador de redes, en concreto en el emulador llamado CORE. Con esto, conseguiríamos el fin último de este proyecto que es realizar pruebas en cualquier tipo de escenario de red ya que el emulador nos permite generar muchos tipos de redes, asignar parámetros físicos en antenas de estaciones base y de dispositivos móviles, simular congestiones de tráfico etc.

I.4 – Estructura de la memoria

La memoria está dividida en secciones, algunas de las cuales, se han subdividido a su vez, con el objetivo de facilitar la búsqueda de información y la lectura por parte de cualquier persona que tenga interés en ello.

- **Estado del arte:** En este capítulo se explican las tecnologías utilizadas para la realización del trabajo y se hace un pequeño estudio comparando las tecnologías utilizadas y las tecnologías descartadas por ser menos apropiadas para la realización del proyecto.

- **Marco Regulador:** En este capítulo se explican la normativa con respecto a las redes móviles IP y su uso.

- **Entorno socioeconómico:** En este capítulo se hace un breve estudio de la posibilidad de implantación del protocolo PMIPv6 en la red de internet y sus beneficios económicos y sociales.

- **Descripción del trabajo realizado:** En este capítulo explicaremos con detalle las tareas que se han realizado para llevar a cabo el proyecto.

- **Conclusiones:** En este capítulo se expondrá una explicación de los resultados obtenidos.

- **Líneas futuras de trabajo:** En este capítulo se expondrán posibles desarrollos adicionales que han quedado fuera del alcance del proyecto pero que podría ser interesantes llevarlos a cabo en el futuro.

- **Planificación de tareas y presupuesto:** En esta parte desglosaremos las tareas en las que se ha dividido el proyecto y explicaremos el presupuesto.

- **Bibliografía:** En este capítulo se detallan las referencias que aparecen a lo largo del trabajo.

- **Anexo1:** Explicación detallada de la configuración del entorno de trabajo

- **Anexo2:** Palabras clave.

- **Anexo3:** Resumen en ingles.

I. INTRODUCTION

I.1 - Generalities

This work bases its interest in the need of mobility without losing internet connection for the mobile devices of the users. With it we search an test enviroment where the execution of the PMIPv6 protocol is easy to analise and besides it can be executed in differents cenaries with the advantage of using only one physical machine.

In this introduction we will explain:

- Motivation
- Objectives
- MemoryStructure
    - State of the art regulatory framework
    - Socioeconomic environment
    - Description of work
    - Conclusions
    - Future lines of work
    - Planning tasks and budget
    - Bibliography
    - Annex 1
    - Annex 2
    - Annex 3

I.2 – Motivation

One of the biggest challenges of technology today is to ensure that all mobile devices are constantly connected to the Internet, avoiding time slots in which the device is offline or getting these spaces as small as possible. This is due to the need for people to use the devices continuously and these increasingly move faster, from cars to airplanes. This problem is easier to solve when the devices move at lows peed, but as this increases, changes occur more frequently and the problem becomes greater as we have many more changes in connection with access points.

On the other hand we would have the problem of the number of connected devices. While a device on a local network can have a private address, they are increasingly connected devices in need of public addresses which must be unique. The appearance of these devices has increased exponentially in recent years and continues to rise, so a system to manage this increase in efficiency is required.

Both problems are solved by PMIPv6, because one side is a protocol specifically designed to solve the problem of mobility and is also based on IPv6 uses 128-bit addresses.
Based on the need for mobility devices in IP networks today, it seems interesting to facilitate testing protocol as PMIPv6. The advantage of this protocol, compared with other proposed for the same purpose, is that there is no need to install additional software on mobile devices, also it is standardized to manage mobility in third generation networks and following, particularly in 3GPP networks [1].

Another motivation is the potential use fullness of this project in teaching, as when executing the protocol in a single machine with an emulator, there source sused are smaller and ease of use, analysis and demonstration of its implementation is much simpler.

I.3 – Objectives

The objective of this project is to get the PMIPv6 protocol emulator work properly in a network, particularly in the emulator called CORE. With this, we would get the ultimate goal of this project is to test any network scenario because the emulator enables us to generate many types of networks, assign physical parameters in base station antennas and mobile devices, simulate traffic congestion etc.

I.4 – Memory Structure

The memory is divided in to sections, some of which are subdivided in turn in order to facilitate the search for information and reading by anyone who has an interest in it.

- **State of the art:** In this chapter the technologies used for the performance of work are explained and a small study done comparing the technologiesused and technologies dismissed as less appropriate for the project.

- **Regulatory Framework:** This chapter explains the rules regarding IP and mobile networks use.

- **Socioeconomic environment:** This chapter provides a brief study of the possibility of implementing the PMIPv6 protocol in the Internet network and its economic and social benefits is made.

- **Description of work done:** This chapter explains in detail the tasks that have been made to carry out the project.

• **Conclusions:** This chapter will provide an explanation of the results obtained.

• **Future lines of work:** In this chapter possible further developments that have been outside the scope of the project but it could be interesting to carry the mount in the future will be presented.

• **Planning tasks and budget:** In this part, we will break down the tasks that have divided the project and explain the budget.

• **Bibliography:** This section are detailed references that appear throughout the work.

• Annex 1: Detailed explanation of the configuration of the work environment.

• Annex 2: Keywords.

• Annex 3: Summary in English.

CAPÍTULO II

ENTORNO SOCIOECONÓMICO

II. ENTORNO SOCIOECONOMICO

Uno de los mayores barómetros del cambio en el marco socioeconómico de las tecnologías en España es el de los SMS.

En España, a finales de 2014 el 80% de los terminales móviles eran de tipo smartphone, esto es, con capacidades de computación y comunicaciones avanzadas. Estas capacidades de computación le permiten hacer uso de aplicaciones de mensajería instantánea, siendo las más populares Whatsapp, Facebook Messenger y Telegram. En el caso de WhatsApp cuenta con un índice de penetración del 70% en España.

Estos nuevos servicios son la mayor competencia del SMS en España, y la prueba es que el número de SMS enviados ha caído un 22% desde 2007, y los ingresos por éstos más de un 27%. Algunas operadoras como Movistar, lanzaron su propia app de mensajería instantánea (Tu Me), como media de reacción, aunque fueron cerradas poco después de su creación, debido a la falta de usuarios.

Este es solo un ejemplo de cómo ha cambiado este entorno en España y si nos fijáramos en el resto de Europa probablemente apreciaríamos la misma tendencia de cambio hacia el uso de las nuevas tecnologías y sustitución de las antiguas por estas, ya que en su mayoría ofrecen los mismos servicios de forma más eficiente y con mejoras considerables.

Debido a este cambio brusco en el uso de las nuevas tecnologías las comisiones de los países se vieron obligadas a regular estos servicios de manera que hubiera una competencia real y los operadores con mayor poder de mercado no aprovecharan esto para eliminar la competencia. Esta regulación es tan importante que se contempla en la Ley General de las Telecomunicaciones (LGTel) [2] en la que según cito textualmente se dice:

La Agenda Digital para Europa, principal instrumento para el cumplimiento de los objetivos de la Estrategia Europa 2020, persigue que para 2020 todos los europeos tengan la posibilidad de acceder a conexiones de banda ancha a una velocidad como mínimo de 30 Mbps, y que, al menos, un 50 % de los hogares europeos estén abonados a conexiones de banda ancha superiores a 100 Mbps. Estos objetivos han quedado incorporados a la agenda digital española, aprobada por el Gobierno en febrero de 2013.

Para ello, según estimaciones de la Comisión Europea, se deberá invertir hasta dicha fecha una cantidad comprendida entre los 180.000 y 270.000 millones de euros. Se calcula que en España serán necesarias inversiones del sector privado por valor de 23.000 millones de euros.

Estas inversiones pueden tener un gran impacto económico y social. La Comisión Europea estima que, por cada aumento de la penetración de la banda ancha en un 10 %, la economía (PIB) crece entre el 1% y el 1,5%. A su vez, la OCDE considera que un incremento del 10% de penetración de banda ancha en cualquier año implica un incremento del 1,5% de la productividad durante los siguientes 5 años.

Asimismo, como ha señalado la Comisión Europea, el despliegue de redes ultrarrápidas puede tener un importante impacto en la creación de empleo, estimándose que la innovación podría generar 2 millones de empleos para 2020, incluidos trabajos en sectores relacionados, como la provisión de contenidos o la fabricación de equipos.

Se ha explicado en este capítulo la importancia socioeconómica de las telecomunicaciones, lo que nos lleva a la necesaria e inevitable regulación de las mismas tal y como se explica en el siguiente capítulo.

CAPÍTULO III

MARCO REGULADOR

III. MARCO REGULADOR

En España el organismo encargado de la regulación del mercado es la Comisión Nacional del Mercado y la Competencia (CNMC).

El Título II de la Ley General de las Telecomunicaciones (LGTel)[2] establece las competencias reguladoras de la CNMC en las cuales se explica que dicho organismo independiente tendrá autoridad para resolver conflictos entre operadores y podrá regular las obligaciones aplicables a los operadores con poder significativo en mercados de referencia.

Citando a Don José María Marín Maqueda, presidente de la CNMC: “Este es un sector sometido a una profunda transformación que exige una adaptación del marco normativo de acuerdo a una visión integral, que conjugue el juego del libre mercado con el respeto de la normativa de competencia y una regulación eficaz.”

El marco regulador se establece a partir de las cuotas de mercado estudiadas en el análisis de los técnicos de la CNMC. Según este análisis se consideró innecesaria la regulación del mercado de telefonía móvil de banda ancha, ya que según los datos obtenidos mostraban una buena competencia en el sector. Esto se debe a la entrada de las Operadoras Móviles Virtuales (OMV) las cuales han equilibrado el mercado en los últimos años quitando poder a operadoras de telefonía más antiguas que tenían la mayoría del mercado copado.

A nivel internacional el estándar regulador de las redes de tercera generación y posteriores es el IMT-2000 [3] aprobado por la ITU, la Unión Internacional de Telecomunicaciones. En este estándar se incluyen las especificaciones del espectro de radio frecuencia, las tarifas y la facturación, la asistencia técnica y los estudios sobre aspectos de reglamentación y política. Por otro lado uno de los principales objetivos de este estándar es la interoperabilidad entre las diferentes tecnologías de comunicaciones inalámbricas que existen.

CAPÍTULO IV

ESTADO DEL ARTE

IV. ESTADO DEL ARTE

En este capítulo se explican las distintas opciones que se han analizado para realizar el trabajo y más en profundidad, las elegidas y las razones para hacerlo.

IV.1 Opciones estudiadas:

IV.1.1 Emulador vs simulador de redes

La diferencia entre ellos es la similitud con la realidad.

Un emulador genera un espacio totalmente igual al real, traduciendo todas las sentencias de código de la máquina generada, a la máquina anfitriona, con las mismas posibilidades y defectos.

Un simulador imita la máquina anfitriona, pero todo lo que se ejecuta en el simulador se hace en el sistema operativo de éste y la máquina anfitriona no influye para nada.

La gran ventaja de un emulador es poder utilizar las implementaciones de protocolos que usamos en sistemas reales sin necesidad de aplicarles modificaciones para su funcionamiento, mientras que un simulador debe adaptarse al entorno de simulación lo que puede hacer distinta la implementación.

IV.1.2 Emulador o distribución en máquinas físicas

Se consideró la posibilidad de ejecutar el entorno en máquinas físicas. Sin embargo, para la realización de la prueba más sencilla posible del protocolo de movilidad, se requiere el uso de cuatro PC's como mínimo. A lo anterior, habría que añadir la necesidad de usar un router o switch de unión entre los PC's. Esta opción es descartada ya que es evidentemente más costosa, complicada de probar, y menos flexible ya que el escenario es siempre el mismo a no ser que se usen más máquinas. Todo ello, se aleja del objetivo de este trabajo que es tener una ejecución sencilla del protocolo para comprobar su funcionamiento.

IV.2 Análisis de herramientas

IV. 2.1 Emulador CORE

CORE [4] es un emulador de redes creado sobre Linux. Es una herramienta basada en una GUI en la que se diseñan las topologías de red de manera muy sencilla, permite añadir nodos como: routers, PC's, hub's o switches y unirlos mediante enlaces para formar escenarios de red. Cuando se ejecutan estos diseños, cada nodo es una maquina virtual ligera de LINUX [5].

Se elige CORE por ser emulador y no simulador; porque ofrece características de movilidad, ya que permite crear redes inalámbricas y tener varios puntos de acceso en una misma red inalámbrica, lo cual es perfecto para que routers a los que se conectan los dispositivos móviles parezcan ser el mismo. A su vez, permite crear cualquier diseño de red, es muy flexible y fácil de utilizar.

CORE es una herramienta capaz de emular redes en una o varias máquinas. Da la posibilidad de conectar varias máquinas físicas a través de él, por lo que también se puede conectar a la red física a la que esté conectado el dispositivo en el que se está ejecutando.

Este programa, fue desarrollado por un grupo de investigadores, especializados en redes, que forman parte de "Boeing Research and Technology division".

A.- Funcionamiento:

Este emulador funciona como una máquina virtual ligera, y puede usarse sobre Linux [5] o sobre FreeBSD[6]. Una de las cosas más importantes de emulador, ya que al crear una maquina virtual para cada nodo se puede ejecutar software en cada uno de los nodos independientemente del resto de nodos. En nuestro caso el software del protocolo de movilidad utilizado está basado en Linux, por lo que se ejecutará CORE sobre una maquina con una distribución de Linux instalada.

B.- Ventajas e inconvenientes:

Ventajas:

- La principal ventaja de CORE es su facilidad de uso ya que tiene una interfaz muy visual a la hora de generar redes, lo cual nos hace muy sencillo la modificación de la red en cualquier momento e incluso para actividades docentes sería muy práctico.

- Es eficiente y con una alta escalabilidad debido a que, al no ser una máquina virtual completa, su carga en el procesador de la maquina anfitriona es muy pequeña por cada nodo introducido en la red, lo que permite el diseño de redes bastante grandes en ordenadores normales, es decir, sin necesidad de que tengan gran potencia de procesamiento o memoria física.

- Por otro lado, una de las ventajas que más nos beneficia, es que CORE ejecuta código de implementaciones para Linux.

- También, aunque menos importante en la realización de este proyecto, permite la conexión del emulador con la red física a la que esté conectada la maquina anfitriona. De tal modo que se podrían conectar dos diseños, creados en CORE, en maquinas diferentes.

- Por último aunque, igual que lo anterior, no tan importante, el código fuente de CORE es fácil de entender y se podría modificar, sin ningún tipo de restricción, en caso de ser necesario (es software libre).

Desventajas: Como todo, CORE tiene sus inconvenientes, aunque no son significativos al nivel de uso que se necesita para la realización del proyecto.

- Tiene una limitación de posibles nodos que se pueden introducir en un escenario; esta limitación se podría decir que es más de la maquina anfitriona que del propio programa ya que al virtualizar las redes, no soporta un número infinito de nodos, aunque sí un número muy grande.

- Un problema que sí podría habernos afectado es que emula la red sobre las capas 3 a 7 [7] lo cual implica que las capas 1 y 2 están siendo simuladas de tal forma que a estos niveles no tenemos una ejecución real y en particular, las conexiones inalámbricas entre nodos podrían afectar a los experimentos realizados. Sin embargo, el software del protocolo de movilidad que hemos probado ha interoperado sin problemas con las redes inalámbricas proporcionadas por OCRE.

- Por último CORE está hecho para trabajar en máquinas básicas como portátiles o PC's de usuarios normales. Por lo tanto, aunque es bastante potente existen programas o máquinas específicas que son más eficientes y reales con la emulación de los escenarios, las cuales permiten un mayor manejo de los parámetros de las redes y la ejecución de escenarios más grandes es más fluida.

C.- Requisitos (instalación)

Como ya hemos dicho con anterioridad, CORE se ejecuta sobre Linux o FreeBSD por lo que una máquina que tenga instalada cualquier distribución de Linux o FreeBSD podría ejecutar CORE. Los requisitos físicos mínimos de la máquina son bastante bajos ya que con un procesador de 2 GHz, una memoria RAM de 2 GB y un espacio en el disco duro de 3 MB, se podría ejecutar. Claro está que, cuanto más potente sea la máquina, mejor y más rápido ejecutará el programa, y más grandes serán las posibles redes que se podrán emular.

D.- Modos de operación

Teniendo en cuenta que nuestro código está desarrollado para trabajar sobre Linux:

- CORE puede ejecutar bajo Linux, utilizando sus espacios de nombres de red, o bajo FreeBSD, usando celdas. [8]

- La recomendación es usarlo bajo Linux ya que es la plataforma a la que más soporte dan los desarrolladores de CORE y por tanto, no hace falta una configuración especial del kernel, además es más sencillo de utilizar.

- La utilización de freeBSD en nuestro proyecto podría dar problemas ya que freeBSD soporta todas las llamadas binarias de Linux, mediante la instalación de un paquete especial, pero no soporta las llamadas específicas de i386, es decir, llamadas al sistema.

E.- Conexiones, tanto físicas como inalámbricas.

El emulador ofrece la posibilidad de usar dos tipos de conexiones en sus escenarios, físicas e inalámbricas. Ambas se pueden llevar a cabo entre cualquiera de los dispositivos que ofrece el programa: routers, host, PC's...

CORE permite configurar parámetros tales como: congestión de la red, potencia de las antenas en conexiones inalámbricas, potencias de recepción, probabilidades de pérdidas de paquetes, etc.

En nuestro trabajo hemos utilizado tanto conexiones inalámbricas como físicas, ya que los nodos en el núcleo de la red van conectados mediante cables de red, y los dispositivos móviles se conectan a los puntos de acceso mediante conexiones inalámbricas.

F.- Cómo simula la movilidad.

Para la simulación de la movilidad únicamente hay que añadir una red inalámbrica a la que se conectaran todos los dispositivos que queramos que tengan acceso a esa red. En nuestro caso conectaremos los puntos de acceso y los dispositivos móviles a la red inalámbrica. Durante la ejecución del escenario será cuando acercando unos dispositivos a otros se crearán conexiones entre ellos, siempre que estén conectados a la misma red inalámbrica.

G.- Servicios que ofrece por defecto.

Con la instalación de CORE, se instalan servicios que se pueden ejecutar o no en los dispositivos. Estos servicios son programas que el CORE utiliza para la ejecución de los protocolos. Estos servicios son por ejemplo: ssh, OSPFv2, OSPFv3, IPForward, DHCP, DefaultRouter. .Los servicios son una forma de establecer, antes del inicio de la ejecución de la emulación de un escenario de red, qué programas se ejecutan en cada dispositivo. Más adelante veremos que CORE permite crear servicios para la ejecución de nuestros propios programas o protocolos en los escenarios creados.

H.- Posibilidad de crear nuevos servicios

Este emulador permite la ejecución de programas en cualquiera de los dispositivos utilizados en los escenarios, pero además permite crear servicios a partir de programas externos a CORE, de tal forma que su ejecución sea más sencilla, tanto en el inicio como en la parada.

IV.2.2 Protocolo PMIPv6

La principal ventaja o característica de PMIPv6 [9] respecto a otros protocolos que también pueden gestionar la movilidad de los dispositivos en redes IP es que no requiere de software adicional en los dispositivos móviles. Otra mejora que ofrece PMIPv6 es la mejora de tiempos de conexión y mejora de tiempos de handover de los dispositivos.

El trabajo está hecho sobre PMIPv6 debido a que es un protocolo relativamente nuevo que, en un futuro próximo, tendrá gran importancia dada su capacidad para gestionar la movilidad. Además, es un protocolo que, al no necesita de software adicional en los nodos móviles, es muy fácil de poner en funcionamiento en una red real ya que solo habría que implantar el código de PMIPv6 en los routers de la red.

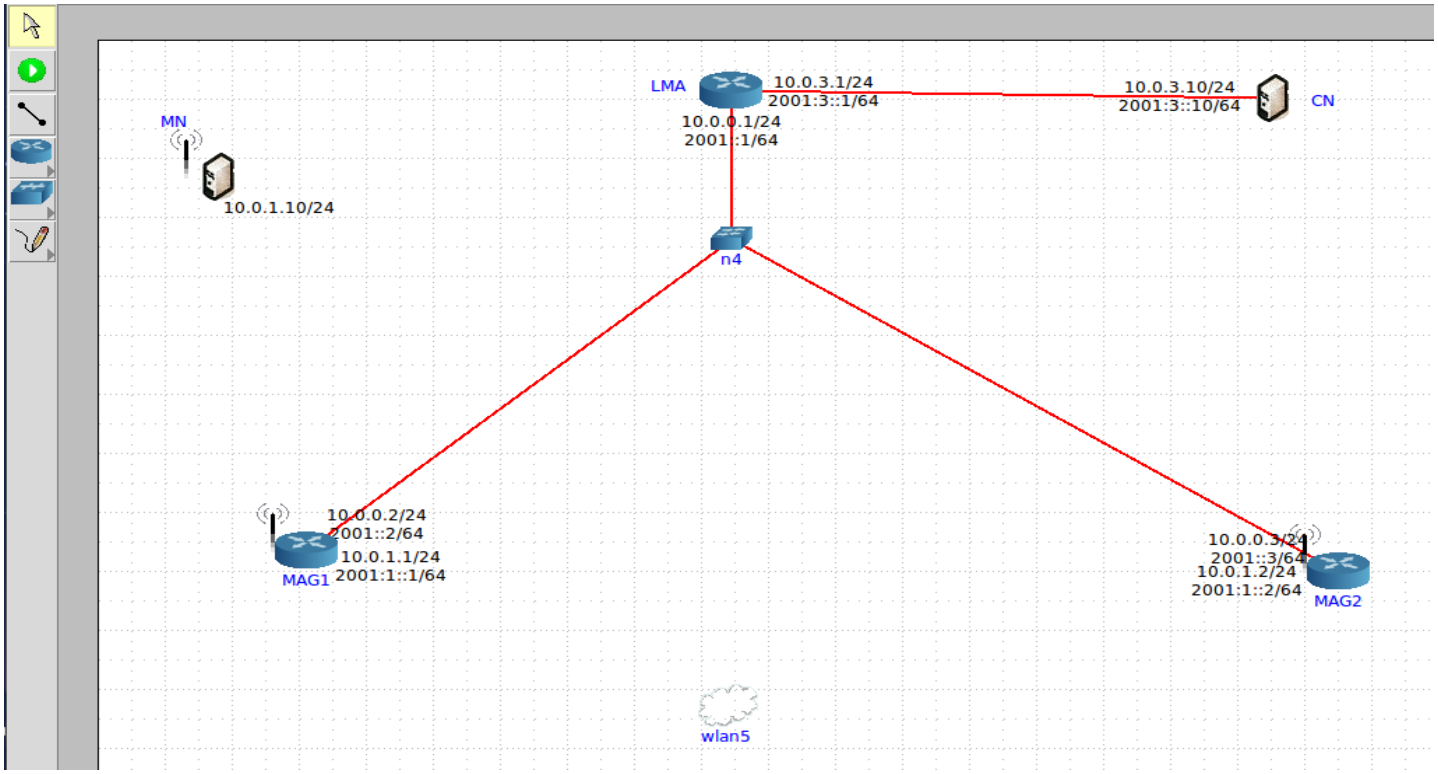


Figura 1: Escenario sencillo

A.- Topología necesaria: Vamos a explicar la terminología utilizada en las redes que usan PMIPv6 y los tipos de nodos necesarios para el funcionamiento de la red.

- **Binding Cache:** Es la tabla en la que el gestor del dominio de PMIPv6 almacena la información de los nodos móviles conectados a su dominio.

- **Binding Cache entry:** Cada una de las filas o entradas de la Binding Cache.

- **Local Mobility Anchor (LMA):** Como su nombre indica, es el ancla topológica de su prefijo de red, es decir, se encarga de gestionar las conexiones a la red de los dispositivos conectados a su dominio y dar acceso a estos al resto de la red.
  Como ya hemos dicho es el responsable de la gestión del dominio, por tanto, este nodo es el encargado de dar direcciones a los nuevos dispositivos que se quieran conectar, almacenando esta información en la Binding Cache.

- **Mobile Access Gateway (MAG): E**s el nodo al que los dispositivos móviles se conectan directamente, por lo tanto, es el encargado de comprobar cuándo hay nodos móviles nuevos queriendo conectarse a la red, y cuándo un nodo móvil se ha desconectado, avisando para todo al LMA. También es el encargado de comprobar si un nodo móvil tiene permiso para conectarse a la red; en caso de no tener permisos, no pasara ningún tipo de mensaje al LMA; podríamos decir que ejerce de filtro para quitar carga de trabajo al LMA.

- **Proxy Mobile IPv6 Domain (PMIPv6-Domain):** Nos referimos al dominio de PMIPv6, hace referencia a la red, que depende de la gestión de un LMA. Esta está compuesta por MAG's y nodos móviles.

- **Mobile Node (MN): U**n nodo móvil como bien dice su nombre, es todo aquel dispositivo móvil, que tiene asignada una dirección IP y cuya movilidad es gestionada por la red. Estos dispositivos, no llevan a cabo ningún tipo de intercambio de mensajes especial, para obtener su dirección IP. Dicho de otro modo, no requieren de un software diferente para conectarse a un dominio PMIPv6.

- **LMA Address (LMAA):** Se denomina así a la dirección IP dada a la interfaz del LMA, que lo une mediante un túnel con el MAG.

- **Proxy Care-of Adres (Proxy-CoA):** Es la dirección asociada a la interfaz de red del MAG que lo une con el LMA.

- **Mobile Node's Home Network Prefix (MN-HNP):** Cada nodo móvil tiene asignado un prefijo en el enlace entre el MN y el MAG. Un MN puede tener varios prefijos para una misma interfaz, en tal caso serán gestionados como bajo una sola sesión PMIPv6.

- **Mobile Node's Home Address (MN-HoA):** Es la dirección que tiene el MN la cual pertenece al MN-HNP y será la utilizada durante toda su conexión a la red, en el dominio PMIPv6.

B.- Funcionamiento: Nos ha parecido interesante insistir en que, para el funcionamiento de PMIPv6, no es necesario ningún tipo de software especial en los dispositivos finales.

Este apartado explica el intercambio de mensajes que se lleva a cabo en el dominio de PMIPv6, durante acciones como el registro de un nuevo MN o el cambio de punto de acceso de un dispositivo o simplemente la desconexión del dominio de un nodo móvil.

Los encargados de gestionar los MN conectados a la red son tanto los MAG´s como el LMA. El primero es el encargado de avisar de nuevas conexiones y desconexiones al LMA y este es el encargado de llevar el registro de todos los MN conectados en su dominio y de que tengan acceso al resto de la red.

A continuación explicamos cómo se realiza la primera conexión de un MN al dominio de PMIPv6.

Cuando un MN entra dentro de un dominio de PMIPv6, debe pedir acceso a un MAG mediante el envío de un Router Solicitation (RS), mensaje incluido en el protocolo IPv6 [10]. El MAG enviará la solicitud, junto con el identificador del MN, al LMA el cual comprobará que el nodo móvil tiene permiso para acceder al dominio [Figura 2].

El mensaje de solicitud que el MAG envía al LMA es un mensaje denominado Proxy Binding Update (PBU) el cual lleva el identificador del MN de tal forma que el LMA pueda comprobar sus permisos de acceso al dominio. En el caso de que estos permisos sean validos el LMA contestara al MAG con un Proxy Binding Acknolegment (PBA) en el cual se incluye el prefijo que se asignará al MN. Por último el MAG enviara un Router Advertisement (RA) en el cual especifica al MN el prefijo que se le ha asignado.

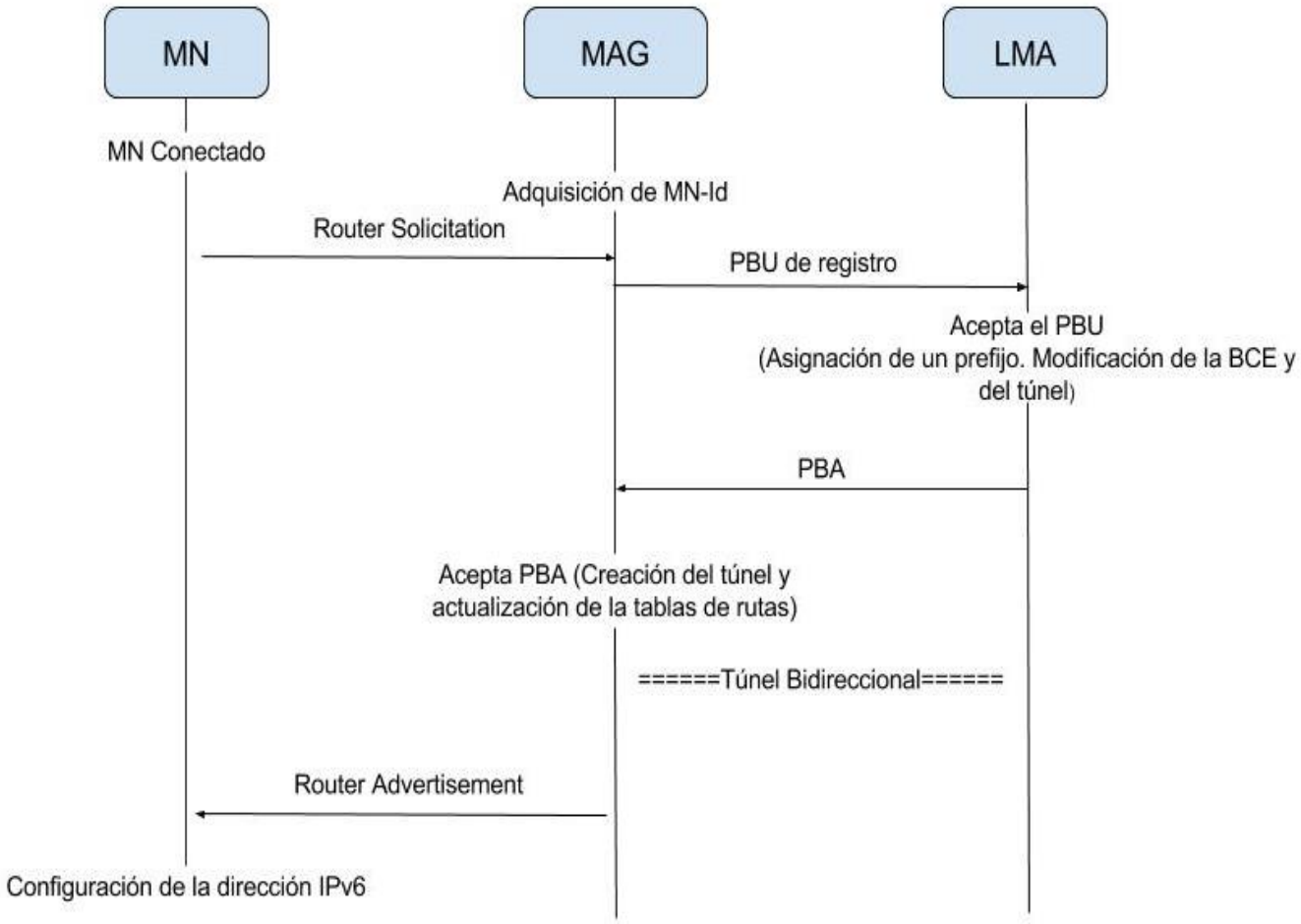


Figura 2: Conexión de un MN nuevo al dominio PMIP.

Cada vez que se conecta un MN nuevo, el MAG y el LMA crean un túnel bidireccional y este lleva asignada, en el lado del MAG, la Proxy-CoA. Igualmente, cada vez que se desconecta un MN, estos túneles son descartados.

El MAG enviará al MN el prefijo que debe asignarse al mismo mediante un RA y este se autoconfigurará una dirección IPv6 global. Una vez terminado esto, el MN pertenecerá al dominio y tendrá acceso al resto de la red.

Dado que es un protocolo creado para gestionar la movilidad de los dispositivos conectados al dominio, los MAG's deberán comprobar con periodicidad que los MN conectados a ellos siguen estando accesibles para que en cualquier caso, avisar de los cambios al LMA. Si un MN cambia de MAG, dentro del mismo dominio de PMIPv6, no cambiará su dirección asignada ya que el LMA lo debe tener registrado de la conexión anterior y le especificara al MAG que le asigne el mismo prefijo de dirección que tenía.

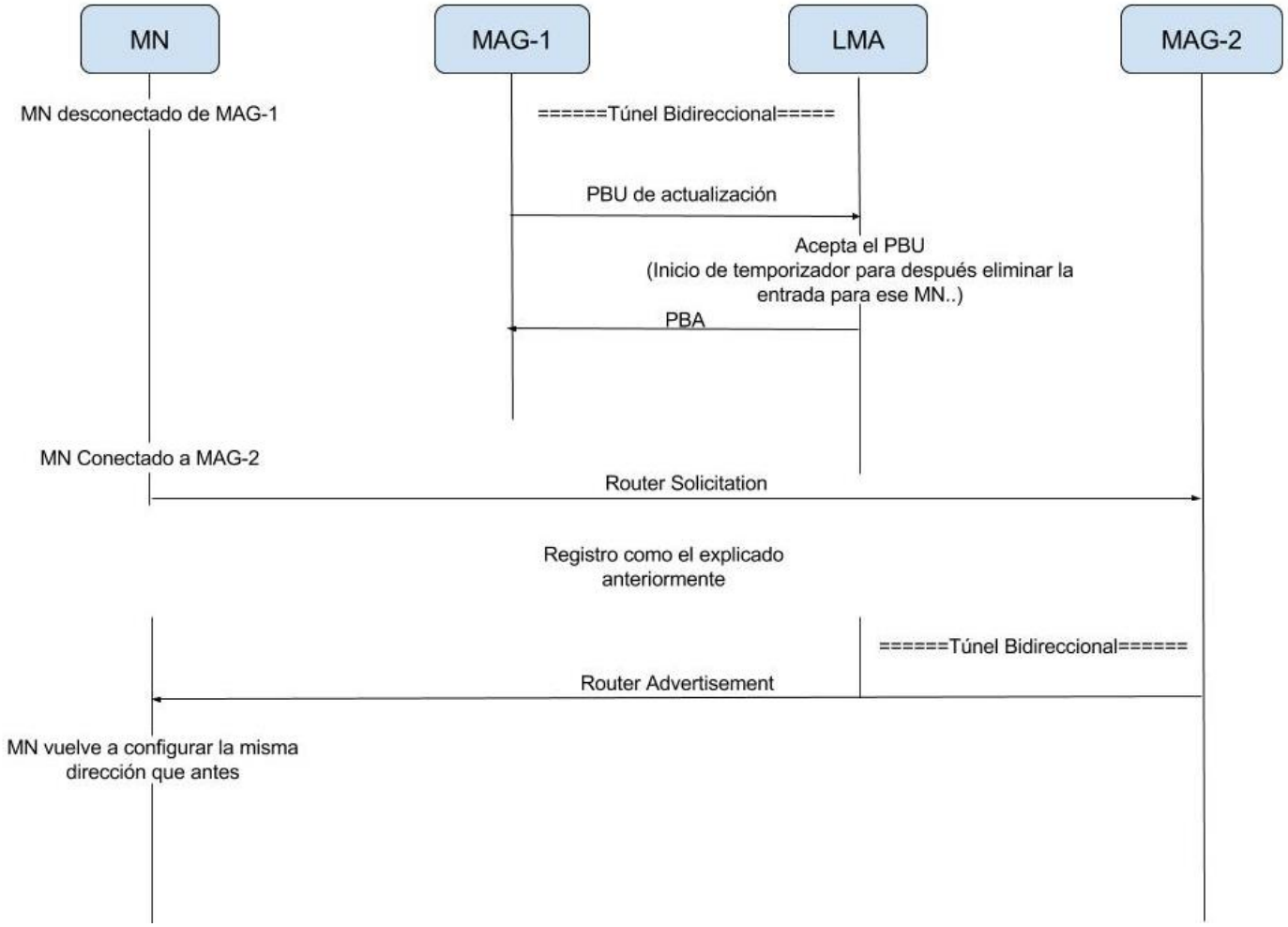


Figura3: MN cambia de MAG dentro del mismo dominio PMIP.

Todos los mensajes como PBU y PBA, se mandan protegidos. Tanto los MAG's como el LMA deben implementar IPsec [11] para asegurar esta protección. Esto es importante debido a que, en estos mensajes, van los identificadores de los MN.

C.- Software utilizado: El código de PMIPv6 utilizado ha sido facilitado por el departamento de Ingeniería Telemática de la Universidad Carlos III de Madrid. Este código está basado en el código creado por un equipo de Eurecom llamado OPENAIRINTERFACE PROXY MOVILE IPv6 (OAI PMIPv6) [12] y posteriormente modificado por investigadores del departamento. Se agradece a Antonio de la Oliva y Carlos J. Bernardos que hayan facilitado el código para la realización del proyecto.

Otro de los programas utilizados ha sido Wireshark [13] el cual se usa para analizar el tráfico total de una red.

D.-Resumen

Una vez repasado el funcionamiento del protocolo PMIPv6 que es el objeto de estudio en este TFG, y estudiado las herramientas empleadas como el emulador CORE en el siguiente capítulo veremos el trabajo realizado.

CAPÍTULO V

DESCRIPCIÓN DEL TRABAJO

V. DESCRIPCIÓN DEL TRABAJO REALIZADO

En este capítulo se explican con detalle las partes del estudio del funcionamiento del código de PMIPv6 en el emulador CORE. Para la realización de este estudio se utilizaron diferentes escenarios y hubo que entender varias cosas sobre la forma de ejecutar el código, además de hacer comprobaciones sobre programas de lectura de mensajes de red y su forma de codificar los mensajes de PMIPv6.

V.1 Preparación del entorno

Esta primera parte es vital a la hora de la buena realización del proyecto dado que con un entorno mal configurado, las pruebas obtenidas podrían no ser fiables e incluso podríamos llegar a pensar que el emulador CORE no permite la ejecución de nuestro código por algún motivo.

La configuración del entorno consiste principalmente en compilar el kernel de Linux, para que tenga activadas todas las opciones de funcionamiento necesarias para la correcta ejecución del código utilizado. También consiste en instalar el emulador CORE y el código.

Sin embargo, al no formar parte esencial del trabajo, he decidido hacer una explicación detallada de los pasos a seguir en el **anexo1** y aquí solo mencionaré los problemas surgidos.

V.2 Puesta en funcionamiento

Lo primero que debemos tener es un escenario en el cual el código pueda ejecutarse con facilidad y nos permita hacer unas primeras pruebas de funcionamiento, de forma sencilla, para comprobar la salida del código. Por tanto, vamos a comenzar por exponer un escenario sacado del emulador que ilustre nuestro escenario principal.

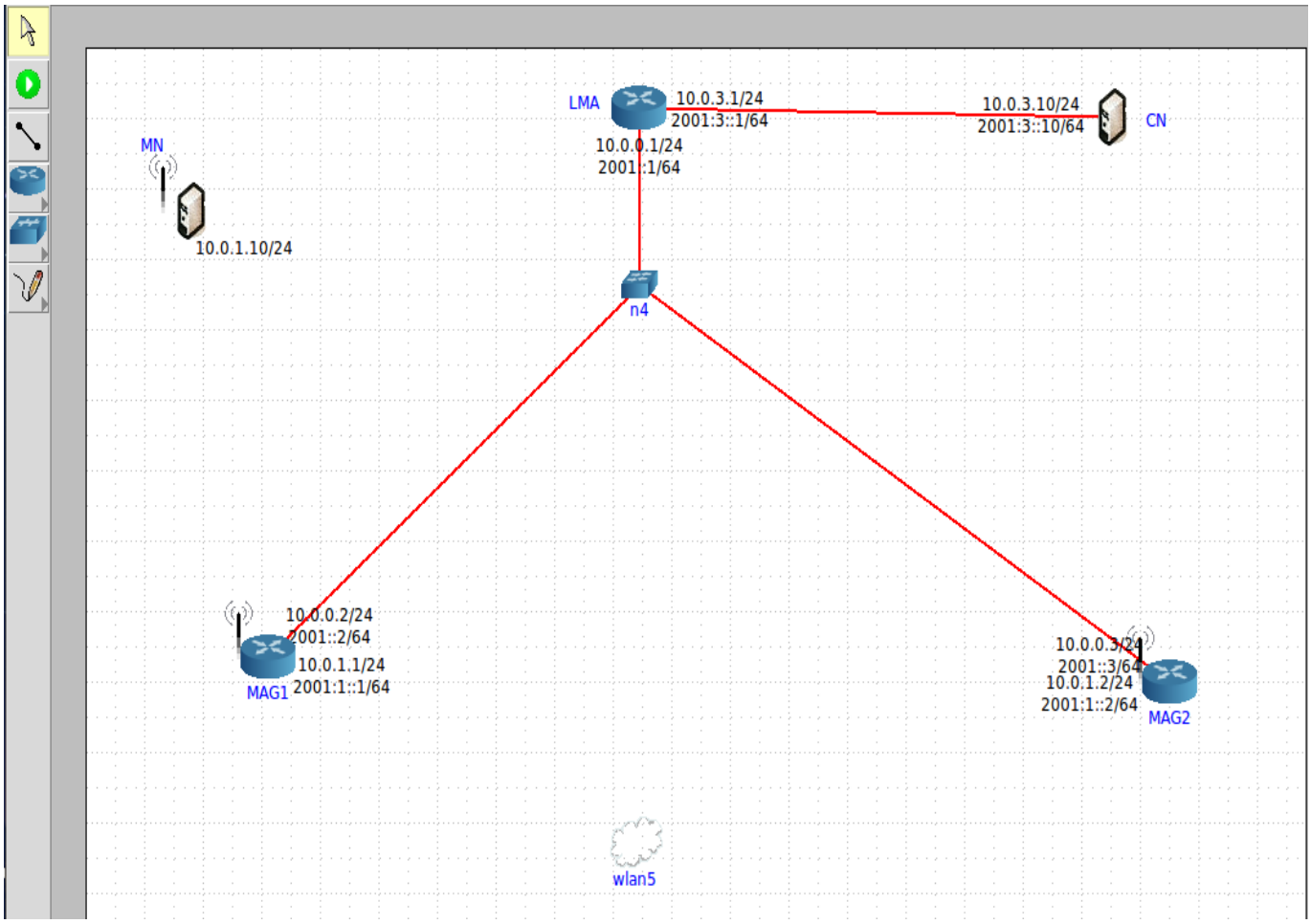


Figura4: Escenario simple de ejecución

a) Creación del escenario:

- Hay que decidir entre las varias posibilidades que existen para hacerlo. Esto se debe a que a la hora de unir los MAG´s, se puede hacer con un router, con un hub, o con un switch. Esta última fue la opción elegida ya que la conexión mediante router necesita de la configuración de las rutas de este para que haya conexión entre los MAG's y el LMA y la conexión mediante un hub implica el envio de paquetes innecesarios por la red. Con el switch nos evitamos tener que configurar las rutas y además conseguimos que los mensajes se envíen solo al destino de estos y no a todos los nodos conectados a él.

- Hay que resolver cómo conseguir que los dos MAG´s muestren el mismo SSID en su red inalámbrica. Esto es bastante fácil ya que, en CORE, no son los MAG´s los que crean una red inalámbrica si no que se crea una red inalámbrica wlan5 y se unen los dispositivos que se quieran conectar. Por tanto, al unir los routers, estos serían los que den acceso a la red a los MN pero, lo harán con las especificaciones puestas en wlan5, con esto y modificando las MAC´s de los interfaces de los MAG´s por las que se unen los MN, conseguimos que los dispositivos que se unan, no noten el movimiento cuando se cambien de MAG.

- Otra de las cosas a configurar correctamente en el escenario consiste en eliminar las direcciones IPv6 que se asignan automáticamente cuando los conectamos con la red inalámbrica. Esto se hace porque los nodos móviles deben configurar su IPv6 al conectarse al dominio PMIPv6.

b) Pruebas iniciales:

Para la realización de las primeras pruebas se usó el escenario descrito en el apartado anterior, además hubo que crear un fichero match. Este fichero es imprescindible para la ejecución los MAG's ya que es en el que se basan para la autentificación de los nodos móviles que se quieren conectar al dominio.

Este fichero contiene una línea por cada nodo móvil que puede conectarse al dominio, en esa línea tendremos dos partes, la primera indica el prefijo a partir del cual el MN configurará su IPv6 y la segunda indica la dirección MAC del nodo al que s le va a permitir el acceso al dominio. El formato de este archivo es muy importante ya que si hay un error el MAG no dará acceso a los MN.

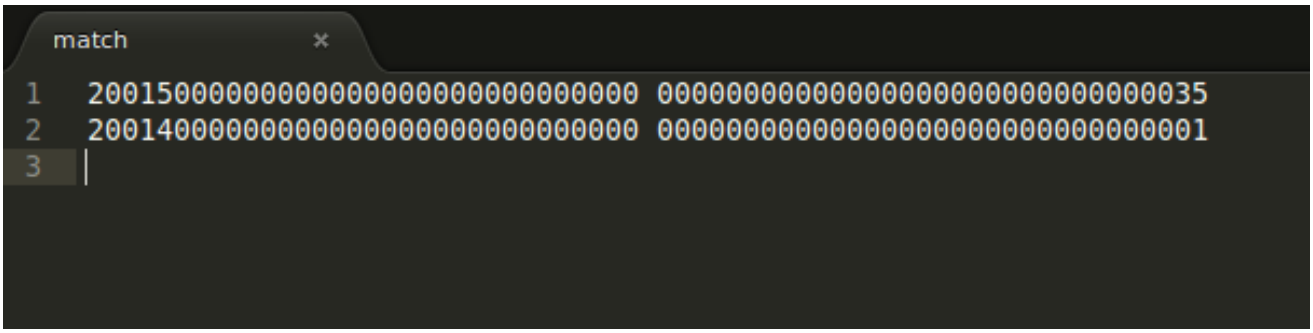


Figura5: Fichero "match"

Una vez configurado el fichero hay que copiarlo a la carpeta /tmp ya que será aquí donde el código de los MAG's comprueba su existencia y en caso de estar comprobará su contenido.

Por último se procede a la ejecución del código en cada uno de los nodos.

Una vez estuvo el escenario ejecutando se procedió a comprobar su funcionamiento.

Se observó que antes de unir el MN a cualquiera de los MAG's estos recibían mensajes de Router Solicitation (RS). Esto se debe al envío de RS por parte de las interfaces virtuales creadas en la máquina anfitriona a los MAG's. Este envío se hace desde las interfaces correspondientes a las interfaces de los MAG's.

Cuando se ejecuta un escenario en CORE, se crea una interfaz virtual en la máquina anfitriona por cada interfaz creada en el emulador. Esto se hace por la posibilidad de CORE de unir los escenarios creados en el programa con la red física a la que esté conectada la máquina anfitriona.

Estas interfaces virtuales llevan una dirección MAC y de red, diferentes a las direcciones correspondientes en el emulador; estas direcciones cambian con cada ejecución del programa.

Esto fue causa de grandes confusiones debido a que en un principio, no se sabía de dónde venían estos mensajes; sin embargo, una vez conocido su origen, fueron útiles ya que cuando llegaban a los MAG's, este prohibía el acceso de estos supuestos dispositivos móviles a la red, por lo tanto, se confirma que el MAG está accediendo bien al fichero "match" y que comprueba si los MN tienen permisos o no.

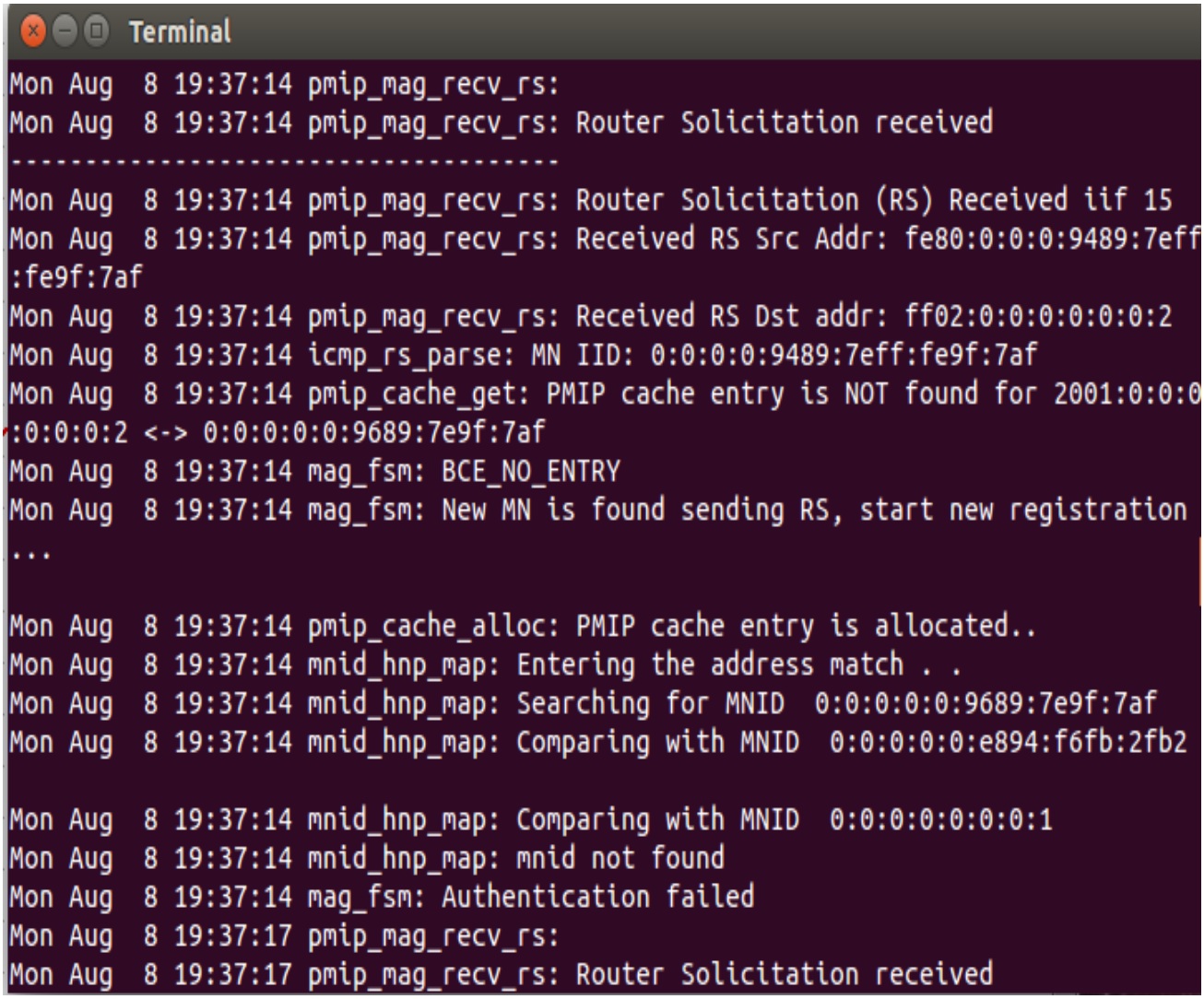


Figura 6: Envío de los RS desde las interfaces virtuales

En esta imagen, vemos en el terminal de un MAG como se reciben RS, desde la dirección origen fe80:0:0:0:9489:7eff:fe9f:7af. Se observa también como el MAG comprueba que no tiene registrado este nodo móvil y como comprueba si éste tiene permisos para conectarse al dominio de PMIP.

19	7.267766000	fe80::ac53:b3ff:fe57:40db	ff02::16	ICMPv6	110 Multicast Listener Report Message v2
20	7.267791000	fe80::ac53:b3ff:fe57:40db	ff02::2	ICMPv6	70 Router Solicitation from ae:53:b3:57:40:db
25	7.848005000	fe80::7812:b3ff:febb:c15d	ff02::16	ICMPv6	110 Multicast Listener Report Message v2
26	7.848014000	fe80::7812:b3ff:febb:c15d	ff02::2	ICMPv6	70 Router Solicitation from 7a:12:b3:bb:c1:5d
32	8.143659000	fe80::ac53:b3ff:fe57:40db	ff02::16	ICMPv6	110 Multicast Listener Report Message v2
38	8.623849000	fe80::7812:b3ff:febb:c15d	ff02::16	ICMPv6	110 Multicast Listener Report Message v2
58	11.279650000	fe80::ac53:b3ff:fe57:40db	ff02::2	ICMPv6	70 Router Solicitation from ae:53:b3:57:40:db
60	11.859940000	fe80::7812:b3ff:febb:c15d	ff02::2	ICMPv6	70 Router Solicitation from 7a:12:b3:bb:c1:5d
80	15.287871000	fe80::ac53:b3ff:fe57:40db	ff02::2	ICMPv6	70 Router Solicitation from ae:53:b3:57:40:db
81	15.867905000	fe80::7812:b3ff:febb:c15d	ff02::2	ICMPv6	70 Router Solicitation from 7a:12:b3:bb:c1:5d

Figura 7: Imagen de wireshark de RS enviados desde las interfaces virtuales

Con la solución del problema anterior lo siguiente era comprobar el mecanismo de conexión del nodo móvil al dominio de PMIPv6.

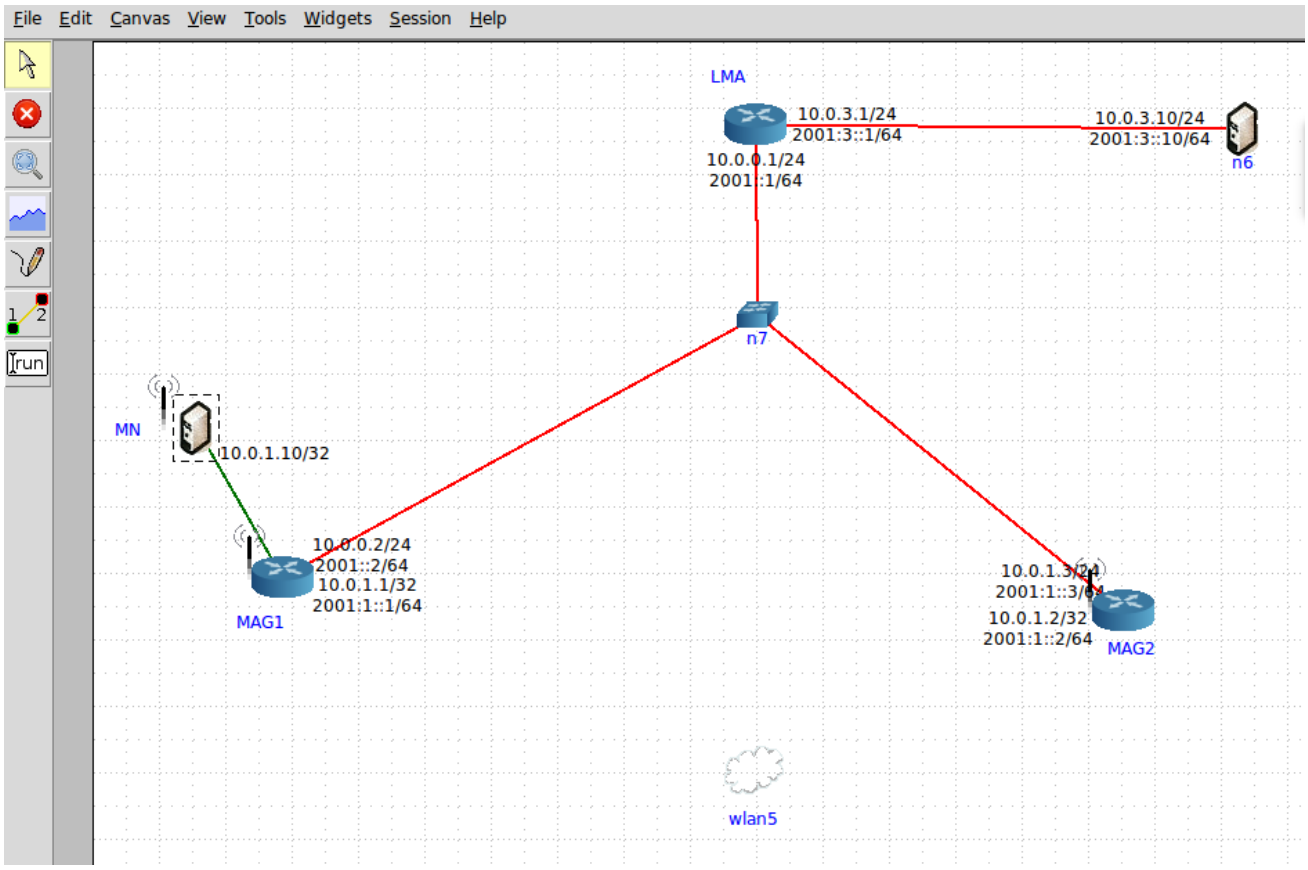


Figura8: Conexión de MN a MAG

Se observo que al conectar el dispositivo móvil al MAG este no recibía ningún RS aparte de los ya recibidos anteriormente. Esto se debe a que los dispositivos móviles del emulador CORE no envían RS por si solos si no que hay que obligarlos mediante el comando “rdisc6”.

Así se consiguió que los nodos móviles obtuvieran acceso al dominio PMIPv6

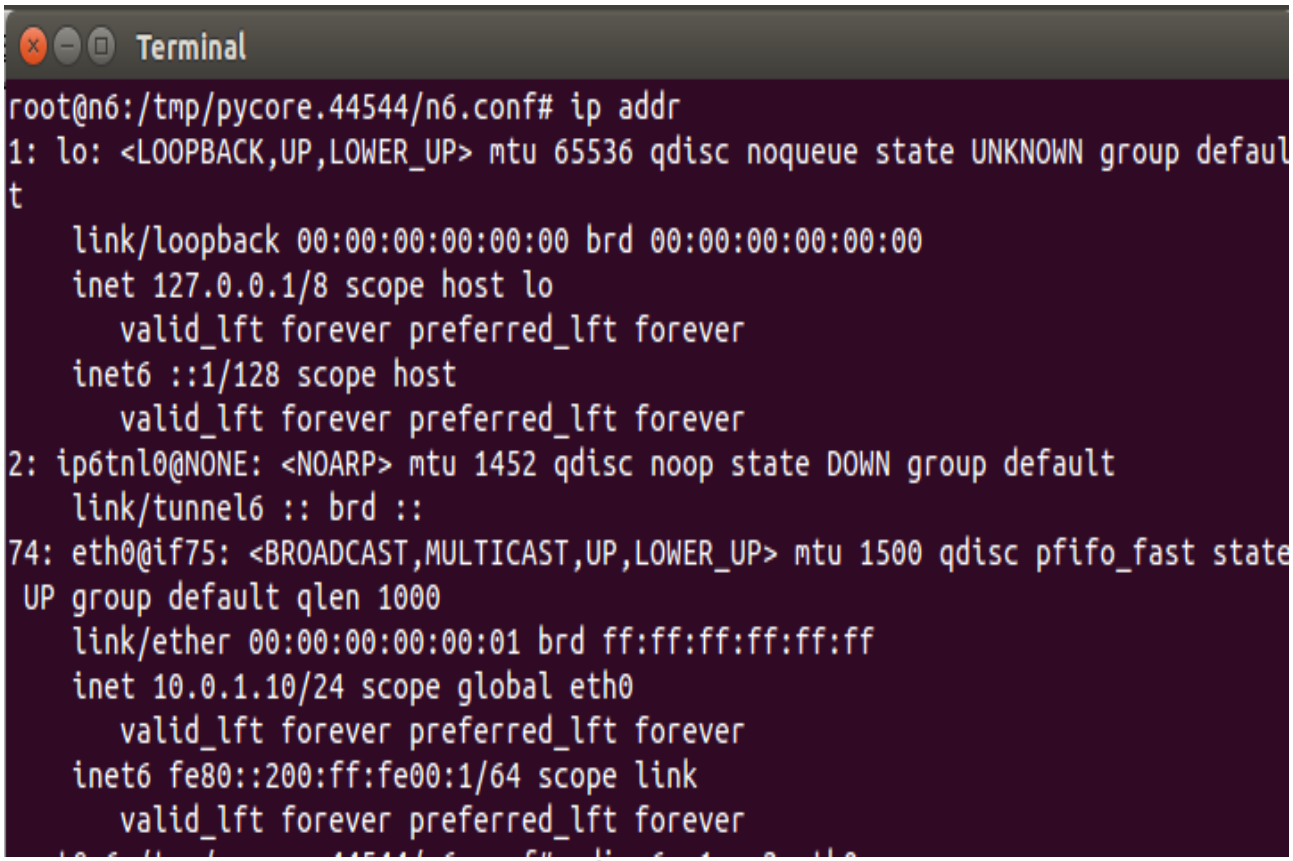


Figura9: MN sin dirección IPv6 global en la interfaz eth0

Una vez conseguido que los nodos móviles enviaran mensajes de Router Olicitation, como parte de las pruebas iniciales de la puesta en funcionamiento, se comprobó que si se mandaba un RS, desde un nodo móvil con una dirección MAC que estuviera incluida en el fichero “match”, el MAG aceptaba la conexión de este al dominio.

```
Thu Sep 22 21:10:49 pmip_mag_recv_rs: Router Solicitation (RS) Received iif 26
Thu Sep 22 21:10:49 pmip_mag_recv_rs: Received RS Src Addr: fe80:0:0:0:200:ff:fe
00:1
Thu Sep 22 21:10:49 pmip_mag_recv_rs: Received RS Dst addr: ff02:0:0:0:0:0:0:2
Thu Sep 22 21:10:49 icmp_rs_parse: MN IID: 0:0:0:0:200:ff:fe00:1
Thu Sep 22 21:10:49 pmip_cache_get: PMIP cache entry is NOT found for 2001:0:0:0
:0:0:0:2 <-> 0:0:0:0:0:0:0:1
Thu Sep 22 21:10:49 mag_fsm: BCE_NO_ENTRY
Thu Sep 22 21:10:49 mag_fsm: New MN is found sending RS, start new registration
...

Thu Sep 22 21:10:49 pmip_cache_alloc: PMIP cache entry is allocated..
Thu Sep 22 21:10:49 mnid_hnp_map: Entering the address match . .
Thu Sep 22 21:10:49 mnid_hnp_map: Searching for MNID  0:0:0:0:0:0:0:1
Thu Sep 22 21:10:49 mnid_hnp_map: Comparing with MNID  0:0:0:0:0:0:0:35
Thu Sep 22 21:10:49 mnid_hnp_map: Comparing with MNID  0:0:0:0:0:0:0:1
Thu Sep 22 21:10:49 mnid_hnp_map: 2001:4000:0:0:0:0:0:0 found the prefix
Thu Sep 22 21:10:49 mag_pmip_md: Making BCE entry in MAG with HN prefix  2001:40
00:0:0:0:0:0:0
Thu Sep 22 21:10:49 mag_pmip_md: New attachment detected! Start Location Registr
ation procedure...
Thu Sep 22 21:10:49 mh_send_pbu: Create PBU with lifetime = 10 seconds (config =
 40 seconds)
Thu Sep 22 21:10:49 mh_send_pbu: Create PBU options...
Thu Sep 22 21:10:49 mh_send_pbu: Send PBU....
Thu Sep 22 21:10:49 pmip_mh_send: Sending MH type 5
from 2001:0:0:0:0:0:0:2
to 2001:0:0:0:0:0:0:1
Thu Sep 22 21:10:49 pmip_mh_send: MH is sent....
Thu Sep 22 21:10:49 mh_send_pbu: Copy PBU message into TEMP PMIP entry iovec....
Thu Sep 22 21:10:49 mag_start_registration: PBU Retransmissions Timer is registe
red....
```

Figura10: MAG recibiendo RS de MN

De esta forma comprobamos que el mensaje RS llega del MN correspondiente y que el MAG lo comprueba con el fichero “match”. Después de esto se puede observar como el MAG inicia el proceso de registro del MN en el dominio mediante el intercambio de mensajes con el LMA.

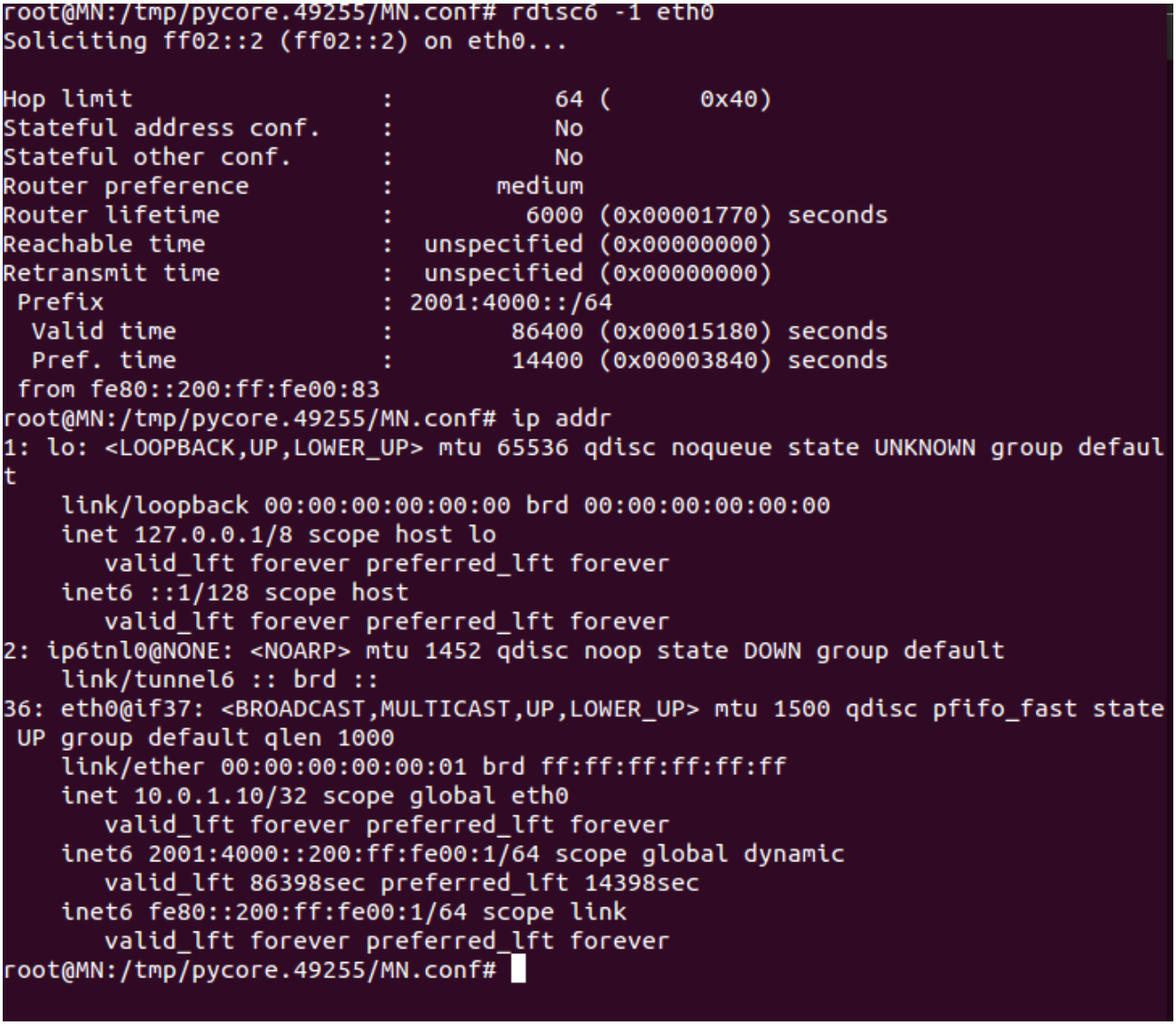


Figura 11: MN al que se le ha asignado un prefijo y ha configurado su dirección IPv6

En esta otra captura observamos cómo después de mandar el comando “rdisc6 -1 -r2 eth0” el nodo con dirección local “fe80::200:ff:fe00:83” le asigna el prefijo de dirección global “2001:1::” al MN. Esta dirección en realidad la propone el MAG al LMA ya que es la que tiene en el fichero “match”, el LMA la acepta y finalmente es el MAG el encargado de enviarla al MN para que este se auto configure la direccionIPv6 basándose en este prefijo y en su dirección MAC. Este no es el correcto funcionamiento de PMIPv6 pero eso se verá más adelante.

c) Pruebas del protocolo y comparativas con la RFC

En esta parte se desarrollaron varias pruebas con el fin de comprobar que el funcionamiento de nuestro código es el explicado en la RFC de PMIPv6.

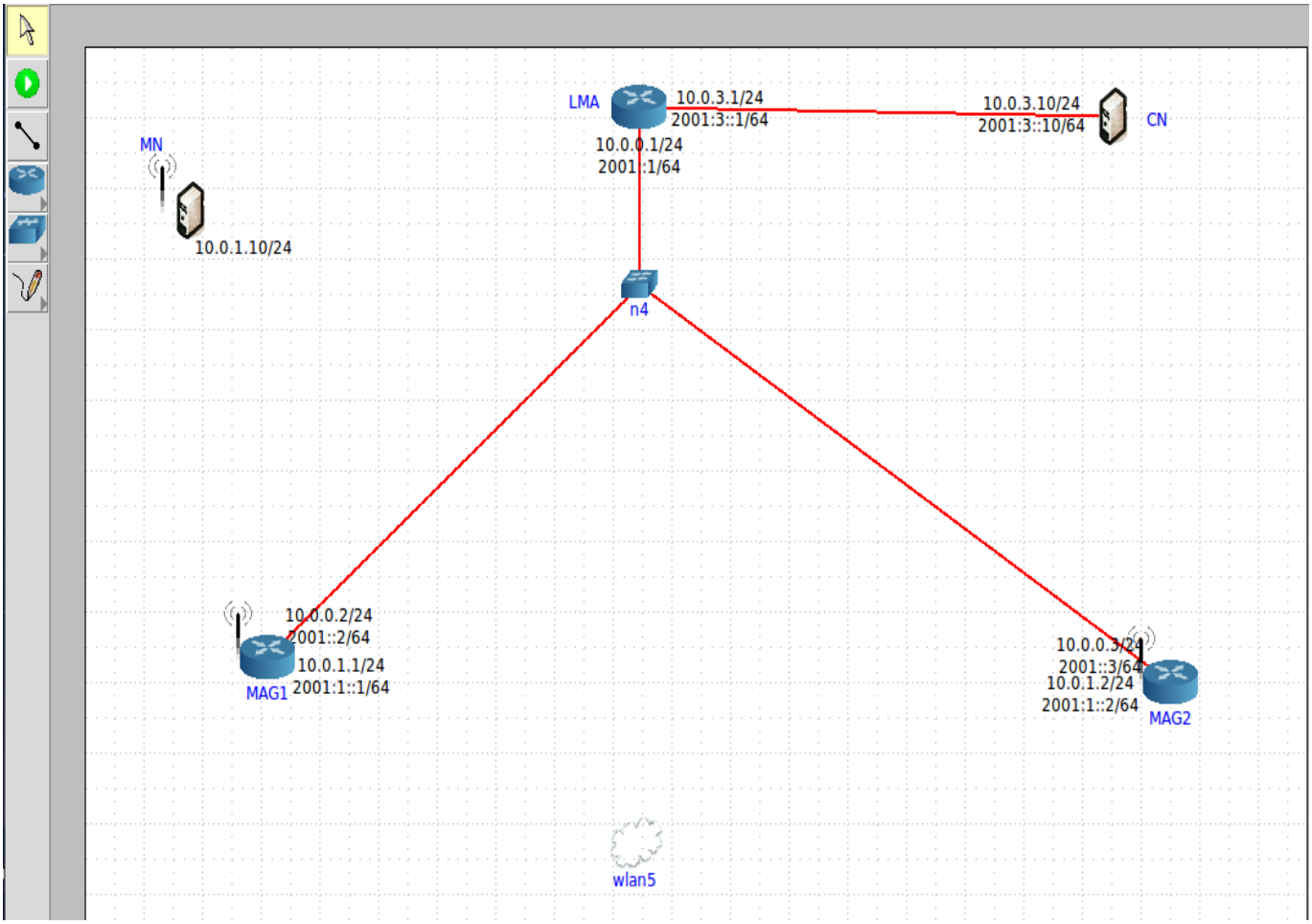


Figura 12: Escenario sencillo

Sobre este escenario configurado de la misma forma que se hizo para las pruebas iniciales se realizaron varias comprobaciones. Todas las comprobaciones se realizaron con el código de PMIPv6 en ejecución.

La primera comprobación realizada consistía en observar si con la unión del primer MN al dominio PMIPv6 se crea un túnel de unión entre el MAG y el LMA. Para ello primero se comprobó que en ningún de estos nodos había creado un túnel antes de la unión del MN.

```
       valid_lft forever preferred_lft forever
2: ip6tnl0@NONE: <NOARP> mtu 1452 qdisc noop state DOWN group default
    link/tunnel6 :: brd ::
26: eth0@if27: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
 UP group default qlen 1000
    link/ether 00:00:00:00:00:83 brd ff:ff:ff:ff:ff:ff
    inet 10.0.1.1/32 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 2001:1::1/64 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::200:ff:fe00:83/64 scope link
       valid_lft forever preferred_lft forever
28: eth1@if29: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
 UP group default qlen 1000
    link/ether 00:00:00:aa:00:00 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.2/24 scope global eth1
       valid_lft forever preferred_lft forever
    inet6 2001::2/64 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::200:ff:feaa:0/64 scope link
       valid_lft forever preferred_lft forever
root@MAG1:/tmp/pycore.49255/MAG1.conf#
```

Figura 13: Terminal del MAG sin túnel creado

Una vez comprobado que los túneles no estaban antes de la unión del MN al dominio, se procedió a la comprobación de la creación de los túneles en el momento de la unión del MN al dominio.

```
Terminal
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ip6tnl0@NONE: <NOARP> mtu 1452 qdisc noop state DOWN group default
    link/tunnel6 :: brd ::
3: ip6tnl1@eth0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1460 qdisc noqueue state UN
KNOWN group default
    link/tunnel6 2001::2 peer 2001::1
    inet6 fe80::ecc4:c1ff:fe7f:78ce/64 scope link
       valid_lft forever preferred_lft forever
72: eth0@if73: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
 UP group default qlen 1000
    link/ether 00:00:00:aa:00:01 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.2/24 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 2001::2/64 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::200:ff:feaa:1/64 scope link
       valid_lft forever preferred_lft forever
76: eth1@if77: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
```

Figura 14: Terminal del MAG con un túnel creado

Se puede observar como en la interfaz eth0 del MAG se crea una interfaz virtual llamada “ip6tnl1@eth0” la cual une las interfaces 2001::2 y 2001::1 que corresponden a la interfaz eth0 y eth0 del MAG y LMA respectivamente.

De esta forma comprobamos que el código se adhería a lo especificado por la RFC 5213.

Para explorar el intercambio de mensajes y después analizar el contenido de cada mensaje se utilizó la herramienta Wireshark versión 1.10.6. Esta herramienta dispone de un filtro para obtener solo los mensajes de las interfaces requeridas, se puede filtrar también por el origen o destino de los mensajes, pero el filtro que más nos interesaba es el que muestra solo los mensajes de un protocolo concreto. Para filtrar los mensajes de PMIPv6 se utiliza el filtro MIPv6, con el cual se podía pensar que la decodificación de los mensajes no iba a ser correcta, sin embargo Wireshark decodifica los mensajes bien aunque los nombre como si fueran mensajes de MIPv6. Esto se comprobó viendo que los mensajes llevaban los flags que añade PMIPv6 y que no aparecen en MIPv6.

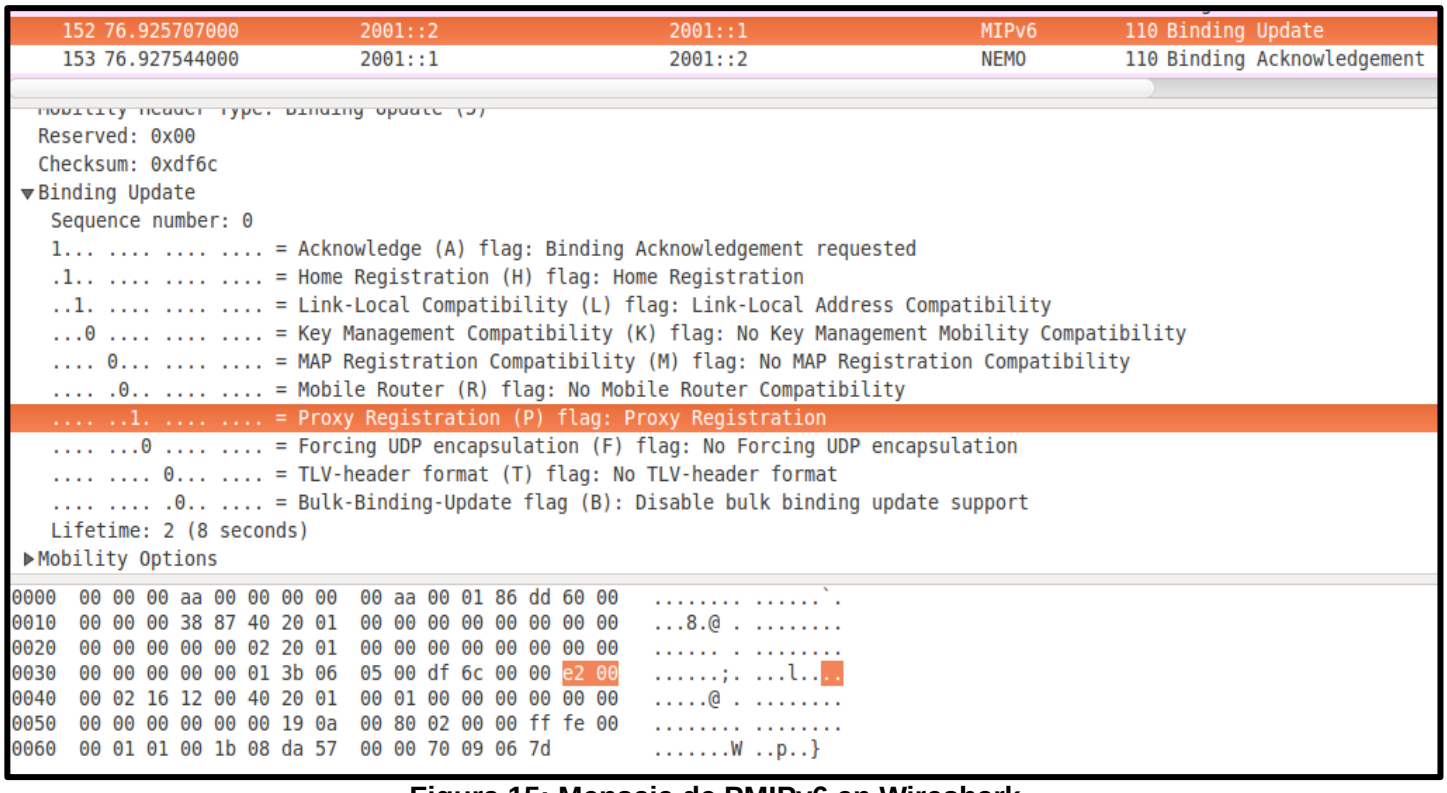


Figura 15: Mensaje de PMIPv6 en Wireshark

Después de asegurarnos que los resultados obtenidos no dependerían de la interpretación de Wireshark pasamos a las pruebas.

Lo primero que comprobamos fue el intercambio de mensajes entre el MAG y el LMA cuando un nodo móvil se conectaba al dominio.

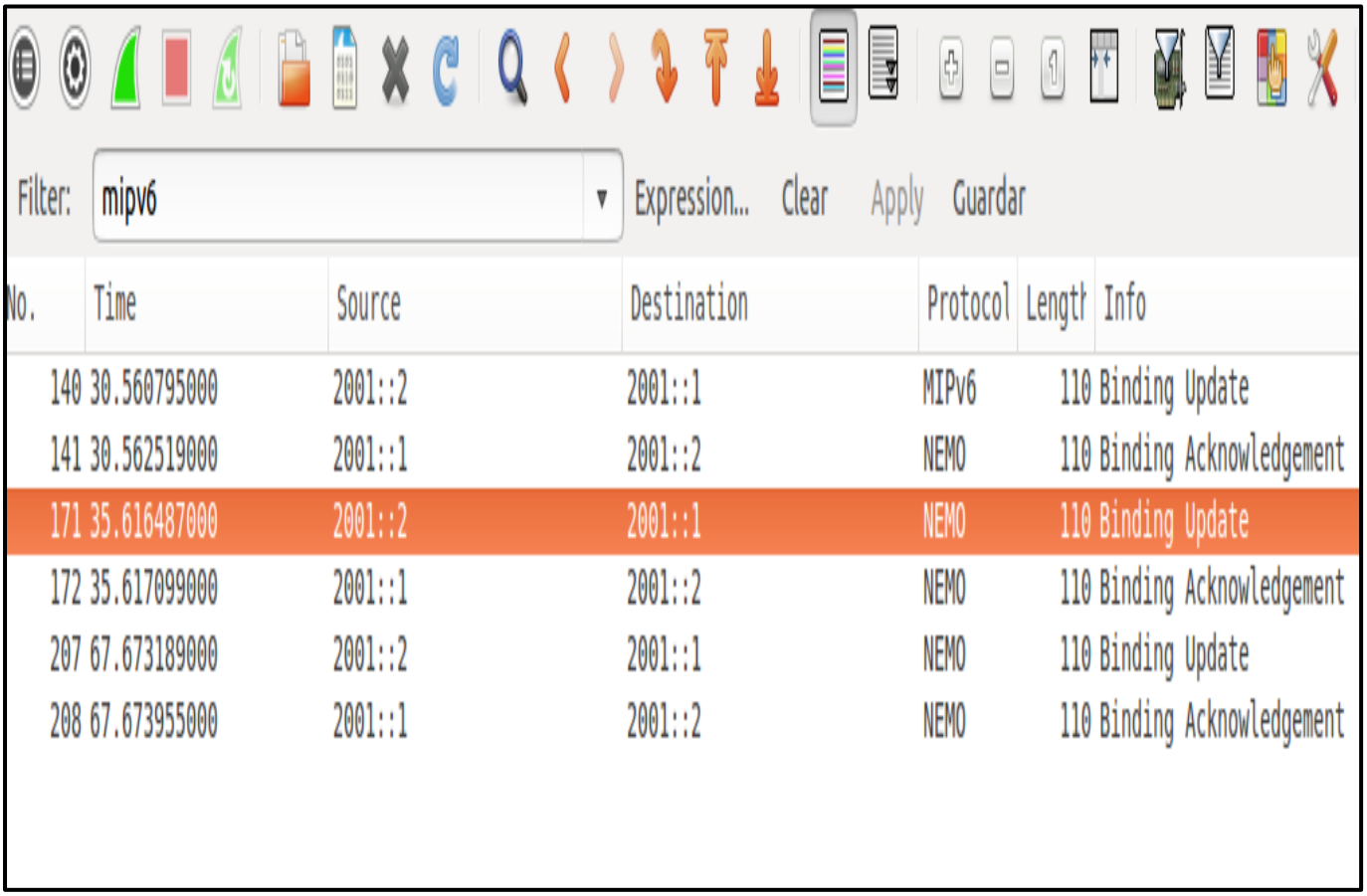


Figura 16: mensajes de conexión entre LMA y MAG para la unión de un nuevo MN

Como vemos en el mensaje resaltado en la imagen anterior, cuyo origen es la dirección IPV6 global del MAG y su destino es la dirección IPv6 global del LMA, el MAG envía un PBU al LMA en el momento en el que se conecta un nodo móvil nuevo al dominio. Se aprecia, como dijimos antes que los mensajes son “Binding Update” y no “Proxy Binding Update” debido a que Wireshark utiliza el nombre genérico para no particularizar a cada protocolo de movilidad.

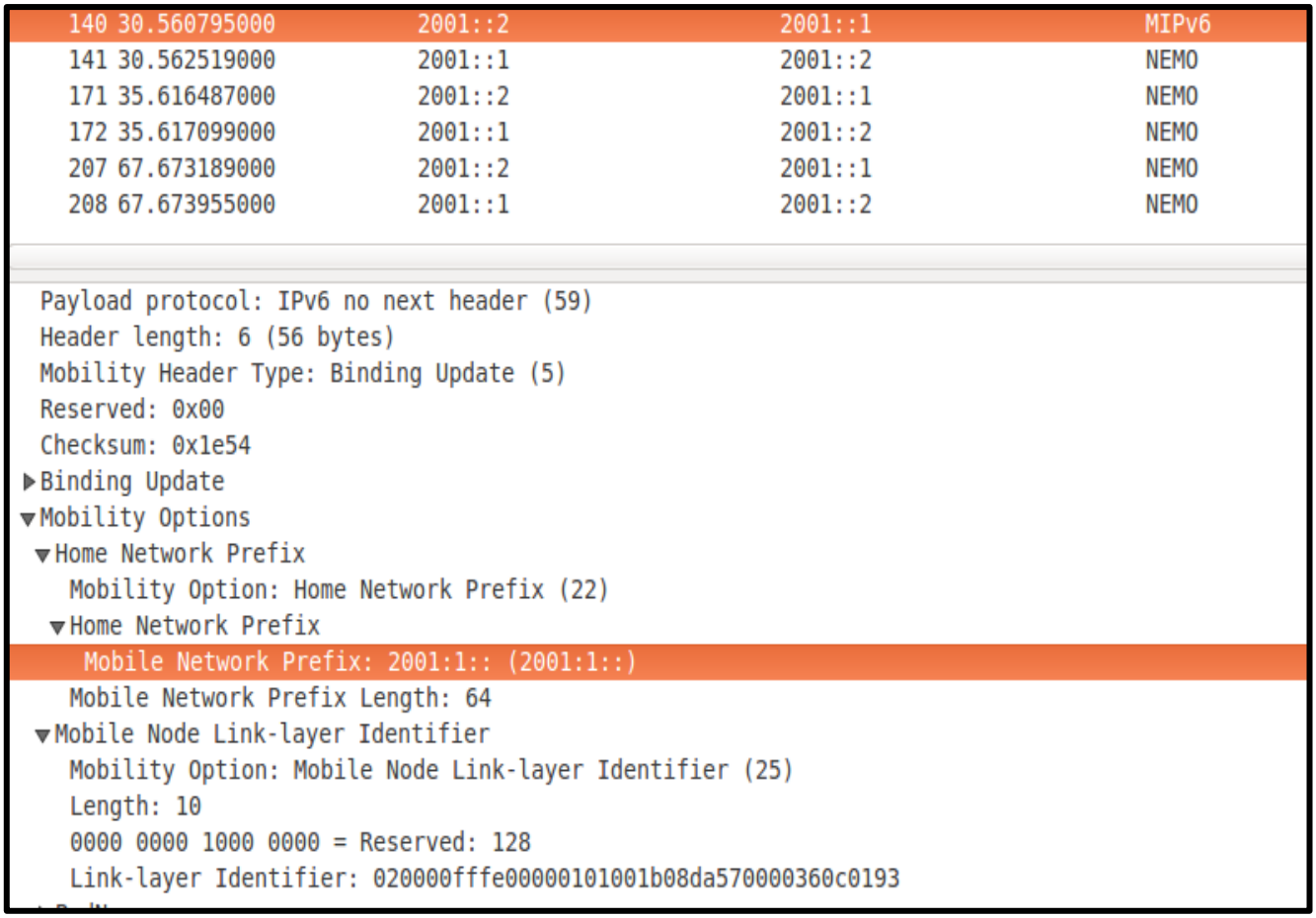


Figura 17: Mensajes, PBU y PBA, de conexión entre LMA y MAG

Como se muestra en la imagen, apreciamos la primera diferencia entre el código utilizado y lo estipulado en la RFC de PMIP. Se observa como el MAG, con dirección IPv6 2001::2, envía al LMA, con dirección IPv6 2001::1, el prefijo que se asignará al MN dentro del contenido del mensaje. Este mensaje no debería contener esta información ya que es el LMA el encargado de asignar un prefijo de red y el MAG no debe influir en esta decisión.

Esta información la envía en el campo “Mobility Options → Home Network Prefix → Home Network Prefix → Mobile Network Prefix. El MAG hace esto como sugerencia y después el LMA decidirá qué dirección asigna, sin embargo el LMA siempre asigna esta dirección.

Después del intercambio de estos mensajes, el MAG tiene que avisar del prefijo asignado al MN mediante del envió de un mensaje de tipo “Router Advertisement” (RA).

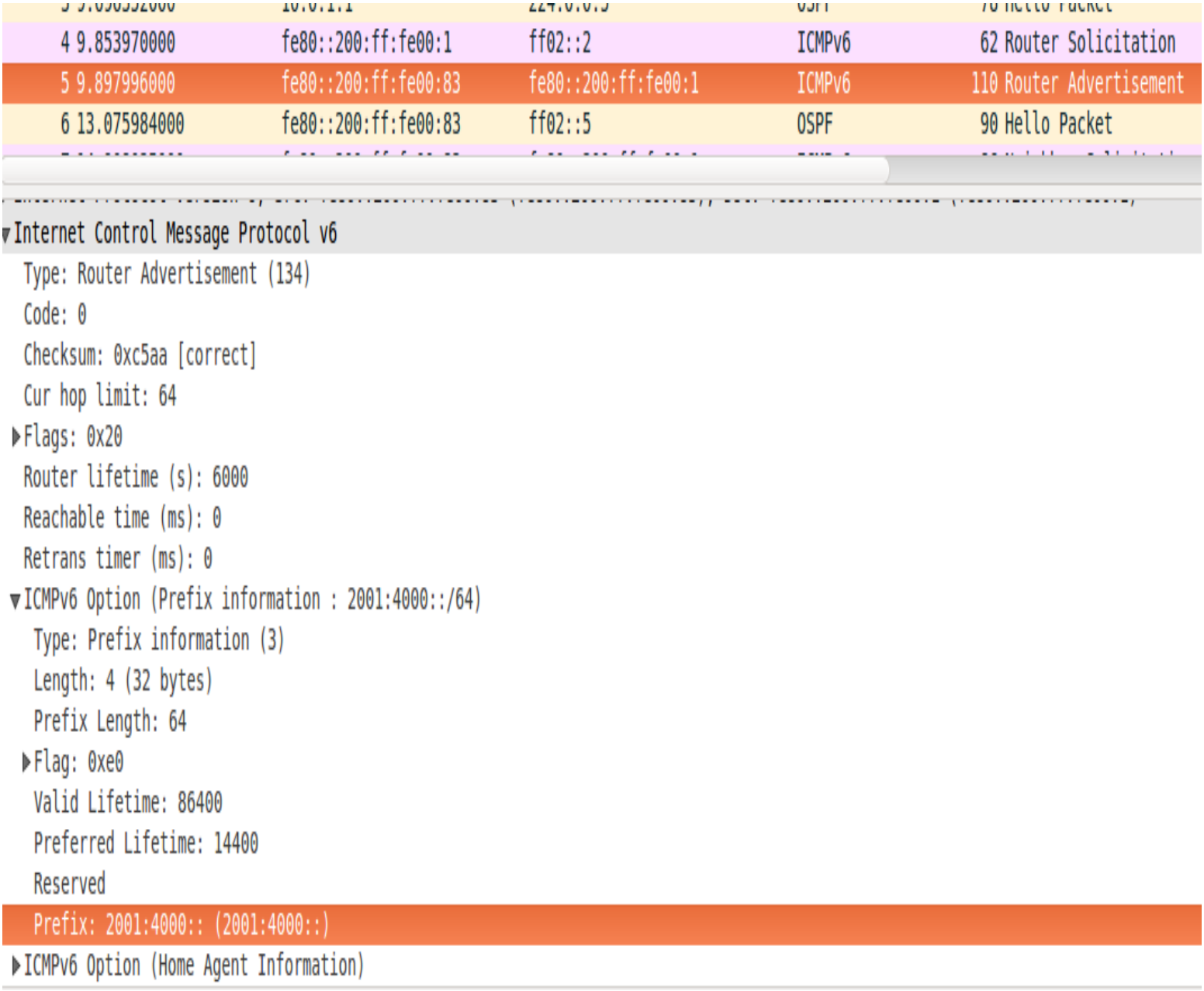


Figura 18: Mensajes entre MAG y MN para asignar prefijo de red

En esta imagen vemos como el MAG con dirección IPv6 link local “fe80::200:ff:feoo:83” envía el prefijo asignado en el mensaje RA. Esta información va en el campo: Internet Control Message Protocol v6 → ICMPv6 Option → Prefix. Este comportamiento si que es el esperado ya que es el mismo que el definido en la RFC de PMIPv6.

Otra de las pruebas que se realizaron fueron las pruebas de “Handoff”, que consiste en comprobar si cuando un nodo móvil se sale del radio de conexión de un MAG este último avisa al LMA.

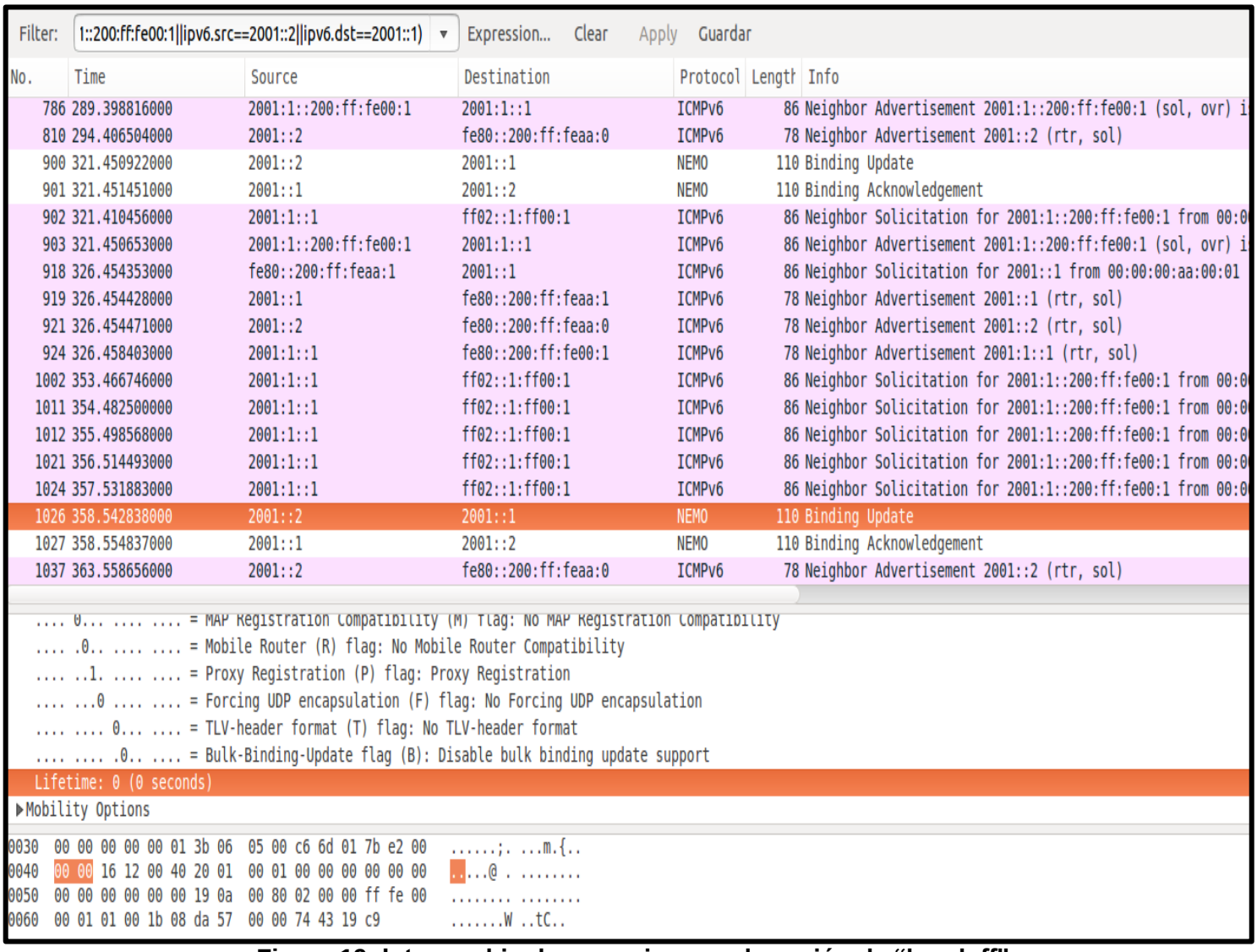


Figura 19: Intercambio de mensajes para la acción de “handoff”

Esta prueba resultó bastante satisfactoria ya que la actuación del MAG es la esperada, es decir, el MAG con dirección IOPv6 global “2001::2” avisa con regularidad al LMA de las conexiones que tiene por parte de los dispositivos móviles para que este actualice su binding cache. Cuando uno se desconecta también tiene que avisar al LMA para que modifique su binding caché. Antes de reenviar el PBU, mensaje que indica que el LMA debe actualizar su binding cache, el MAG se asegura de que sus nodos siguen siendo accesibles a través de él. Como vemos en la imagen anterior, esto lo hace mediante el envío de un NS, hacia la red inalámbrica por su interfaz con dirección IPv6 global 2001:1::1, a una dirección de red IPv6 de tipo **multicast Solicited-node** “ff02::1:ff00:1” a la cual

pertenecen las IPv6 de los nodos móviles; Este mensaje lo envía varias veces por si hubiera congestión en la red y cuando no recibe una respuesta envía al LMA un PBU con la opción “Lifetime a 0”.

Esta prueba se realizó también con más nodos móviles para comprobar que en el envío de PBU iba la información solo del nodo desconectado y que los demás los mantenía. Igualmente se obtuvo el resultado esperado, es decir, el estandarizado por la RFC. Solo hay que eliminar la entrada del nodo que se sale del dominio manteniendo al resto sin pérdida de conexión en ningún instante.

V.3 Pruebas finales

Una vez que se comprobó que los mensajes de señalización se enviaban con la información y en el orden descrito en la RFC 5213, se procedió a hacer pruebas de funcionalidad.
En esta parte se hicieron pruebas como:

Lanzar un ping entre un nodo móvil y un nodo corresponsal y comprobar la conexión entre ambos, cambiar el nodo móvil de MAG y ver cuánto tardaban los paquetes en volver a llegar a su destino y comprobar la conexión entre nodos conectados al mimo dominio tanto si estaban conectados al mismo MAG como si lo estaban en MAG’s diferentes. Estos datos no son del todo fiables debido a la necesidad de los nodos móviles de ser obligados mediante un comando, a mandar un RS a los puntos de acceso para obtener el prefijo IPv6. El resultado ha sido bueno ya que no se perdían paquetes.

Prueba de conexión entre un MN y un CN.

En el escenario utilizado para la realización de estas pruebas se utilizaron dos nodos móviles y un nodo corresponsal el cual actúa como el resto de la red.

El fichero “match” utilizado es el siguiente en el cual cambia con respecto a la versión anterior el prefijo del nodo con dirección MAC 00:00:00:00:00:35 ya que de usar un prefijo 2001:5000/64 causaría problemas dado que es el prefijo utilizado por la red inalámbrica.

```
match
1  2001600000000000000000000000000 00000000000000000000000000000035
2  2001400000000000000000000000000 00000000000000000000000000000001
3
4
```

Figura 20: Fichero match

Para esta primera prueba solo se utilizaron el nodo n4 y en nodo n6 ya que si el nodo n4 tiene conexión con el resto de la red cabe esperar que el nodo n8 haga lo mismo.

Para la configuración de este escenario se necesita:

- Cambiar las direcciones MAC de la interfaces de los MAG's que dan acceso al dominio PMIP a los MN de tal forma que las dos tengan la misma dirección.
- Eliminar las direcciones IPv6 que se asignan automáticamente a los MN al conectarlos a la red inalámbrica.
- Poner direcciones Mac fijas a los MN de tal forma que sean las mismas que aparecen el fichero "match".

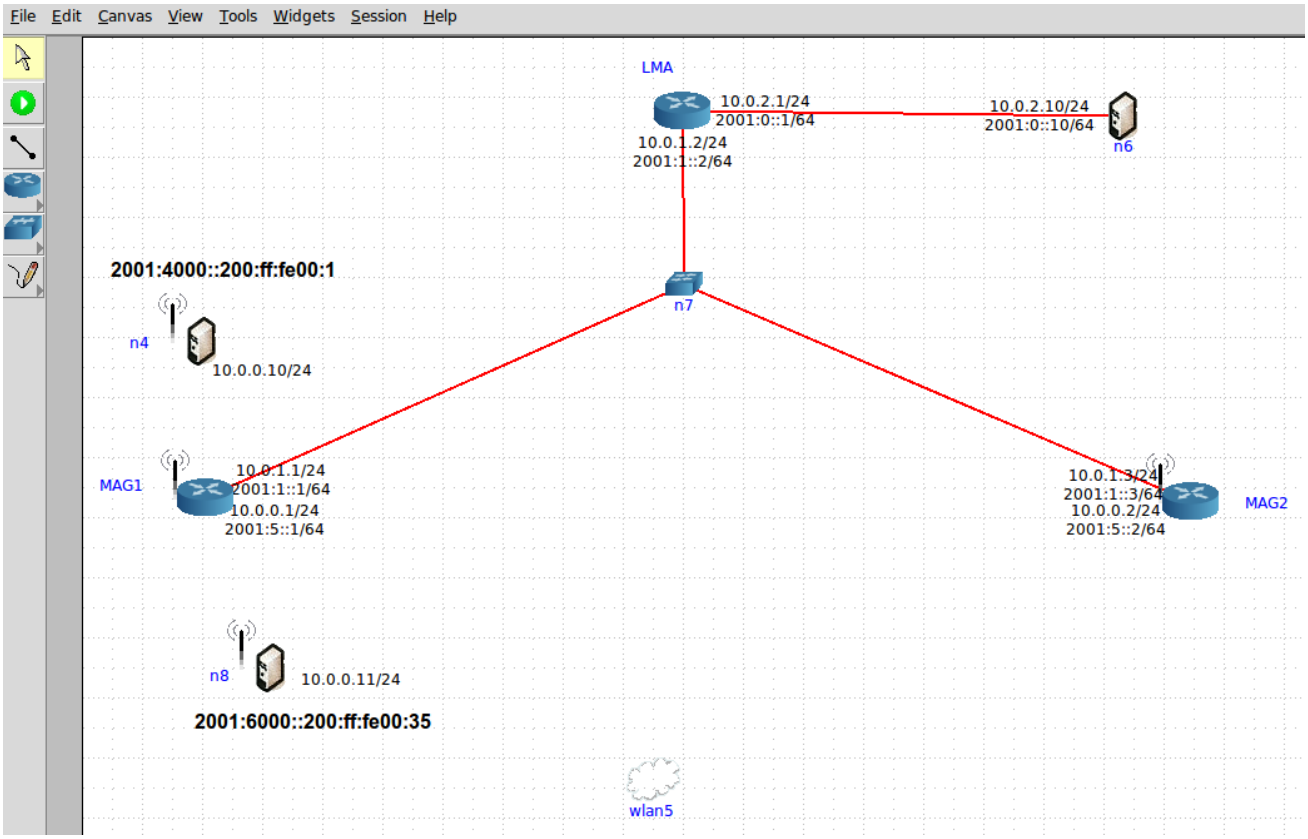


Figura 21: Escenario sencillo 2

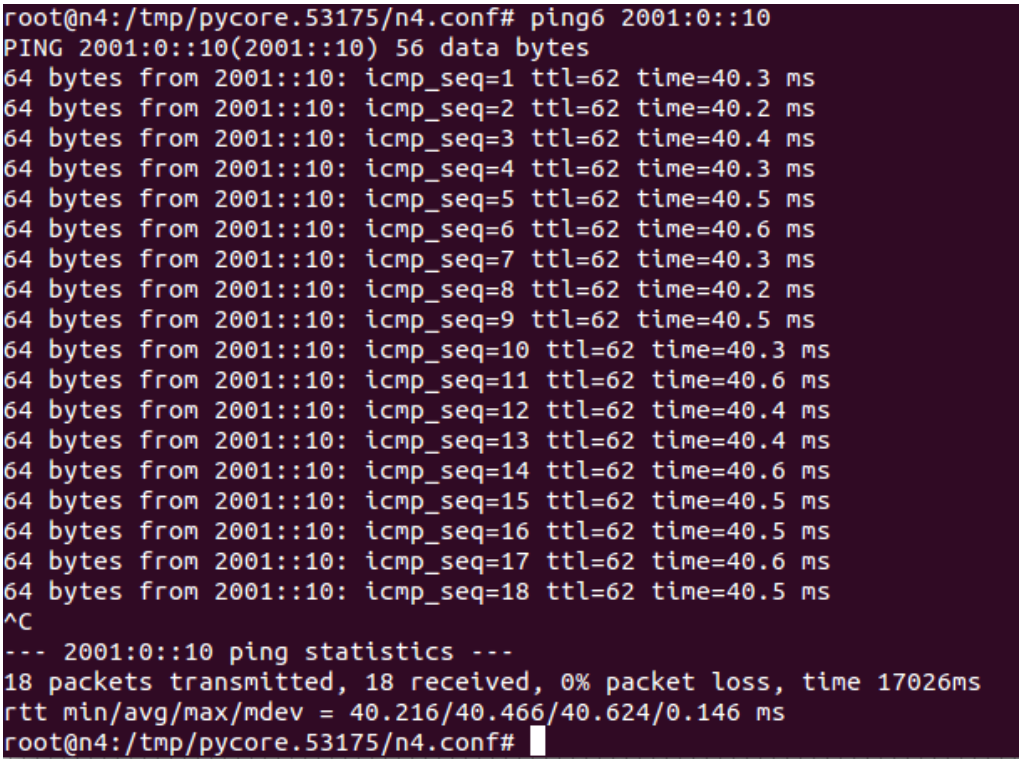


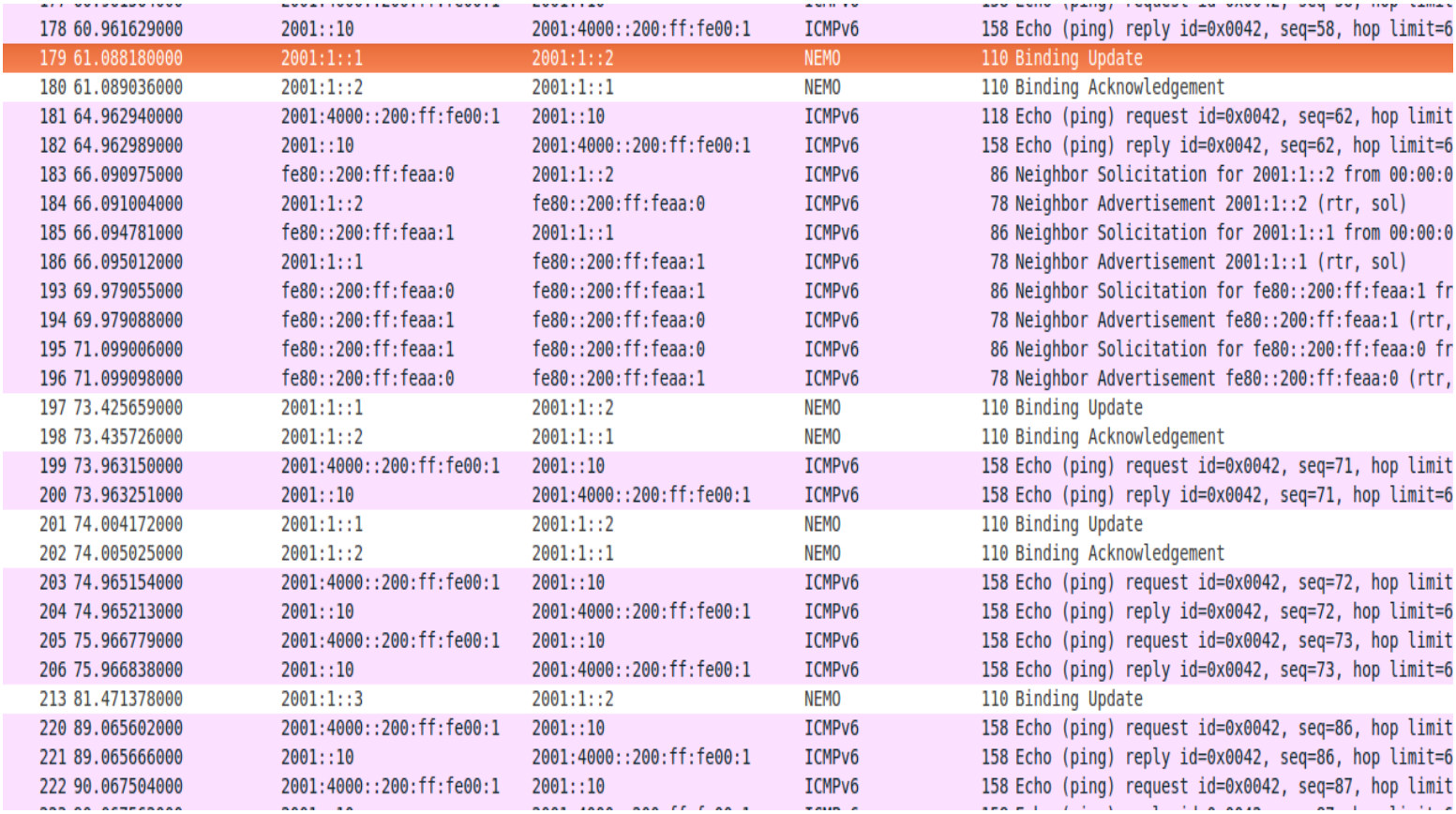
Figura 22: Ping entre MN y CN

En esta captura del terminal del MN vemos como la conexión ente el nodo móvil, que está dentro del dominio de PMIPv6 y el nodo corresponsal, que estaría en cualquier punto de la red, es una conexión continua y sin cortes.

En algunas ocasiones puede darse la perdida de algunos paquetes debido a la señalización entre los MAG's y el MN. Esto es más un problema de soporte por parte de la maquina anfitriona que un problema causado por el protocolo PMIPv6.

Conexión entre MN y CN cambiando el MN de MAG.

El escenario utilizado es el mismo que el descrito en la prueba de conexión entre MN y CN, con la única diferencia que en este caso se hará uso de los dos MAG's



178	60.961629000	2001::10	2001:4000::200:ff:fe00:1	ICMPv6	158 Echo (ping) reply id=0x0042, seq=58, hop limit=6
179	61.088180000	2001:1::1	2001:1::2	NEMO	110 Binding Update
180	61.089036000	2001:1::2	2001:1::1	NEMO	110 Binding Acknowledgement
181	64.962940000	2001:4000::200:ff:fe00:1	2001::10	ICMPv6	118 Echo (ping) request id=0x0042, seq=62, hop limit
182	64.962989000	2001::10	2001:4000::200:ff:fe00:1	ICMPv6	158 Echo (ping) reply id=0x0042, seq=62, hop limit=6
183	66.090975000	fe80::200:ff:feaa:0	2001:1::2	ICMPv6	86 Neighbor Solicitation for 2001:1::2 from 00:00:0
184	66.091004000	2001:1::2	fe80::200:ff:feaa:0	ICMPv6	78 Neighbor Advertisement 2001:1::2 (rtr, sol)
185	66.094781000	fe80::200:ff:feaa:1	2001:1::1	ICMPv6	86 Neighbor Solicitation for 2001:1::1 from 00:00:0
186	66.095012000	2001:1::1	fe80::200:ff:feaa:1	ICMPv6	78 Neighbor Advertisement 2001:1::1 (rtr, sol)
193	69.979055000	fe80::200:ff:feaa:0	fe80::200:ff:feaa:1	ICMPv6	86 Neighbor Solicitation for fe80::200:ff:feaa:1 fr
194	69.979088000	fe80::200:ff:feaa:1	fe80::200:ff:feaa:0	ICMPv6	78 Neighbor Advertisement fe80::200:ff:feaa:1 (rtr,
195	71.099006000	fe80::200:ff:feaa:1	fe80::200:ff:feaa:0	ICMPv6	86 Neighbor Solicitation for fe80::200:ff:feaa:0 fr
196	71.099098000	fe80::200:ff:feaa:0	fe80::200:ff:feaa:1	ICMPv6	78 Neighbor Advertisement fe80::200:ff:feaa:0 (rtr,
197	73.425659000	2001:1::1	2001:1::2	NEMO	110 Binding Update
198	73.435726000	2001:1::2	2001:1::1	NEMO	110 Binding Acknowledgement
199	73.963150000	2001:4000::200:ff:fe00:1	2001::10	ICMPv6	158 Echo (ping) request id=0x0042, seq=71, hop limit
200	73.963251000	2001::10	2001:4000::200:ff:fe00:1	ICMPv6	158 Echo (ping) reply id=0x0042, seq=71, hop limit=6
201	74.004172000	2001:1::1	2001:1::2	NEMO	110 Binding Update
202	74.005025000	2001:1::2	2001:1::1	NEMO	110 Binding Acknowledgement
203	74.965154000	2001:4000::200:ff:fe00:1	2001::10	ICMPv6	158 Echo (ping) request id=0x0042, seq=72, hop limit
204	74.965213000	2001::10	2001:4000::200:ff:fe00:1	ICMPv6	158 Echo (ping) reply id=0x0042, seq=72, hop limit=6
205	75.966779000	2001:4000::200:ff:fe00:1	2001::10	ICMPv6	158 Echo (ping) request id=0x0042, seq=73, hop limit
206	75.966838000	2001::10	2001:4000::200:ff:fe00:1	ICMPv6	158 Echo (ping) reply id=0x0042, seq=73, hop limit=6
213	81.471378000	2001:1::3	2001:1::2	NEMO	110 Binding Update
220	89.065602000	2001:4000::200:ff:fe00:1	2001::10	ICMPv6	158 Echo (ping) request id=0x0042, seq=86, hop limit
221	89.065666000	2001::10	2001:4000::200:ff:fe00:1	ICMPv6	158 Echo (ping) reply id=0x0042, seq=86, hop limit=6
222	90.067504000	2001:4000::200:ff:fe00:1	2001::10	ICMPv6	158 Echo (ping) request id=0x0042, seq=87, hop limit

Figura 23: Cambio de MAG en Wireshark

Esta captura de Wireshark está realizada en la interfaz eth0 del LMA de tal forma que se ven tanto los mensajes de ping como los mensajes de señalización del protocolo PMIPv6. Observamos como el ping ejecutado entre el MN y el CN es interrumpido en el momento en el que el MN sale de la zona de cobertura del MAG2. El ping vuelve a ejecutarse en el momento que el MN se une al MAG1, mensaje BU marcado en naranja en el que el MAG1 indica al LMA que se ha conectado un MN nuevo. Se observa una pequeña perdida de paquetes debido a la señalización entre el LMA y el MAG1. El último BU que observamos con dirección origen 2001:1::3 es el de señalización del MAG2 al LMA en el que el Lifetime se pone a "0" para indicar la perdida de conexión del MN por parte del MAG2 al LMA.

Conexión de MN conectados al mismo dominio.

Utilizando el mismo escenario y configurado de la misma manera que en las dos pruebas anteriores, se procedió a comprobar si había fallos en la conexión cuando los dos nodos se encontraban dentro del mismo dominio.

Conexión de los MN en MAG's diferentes.

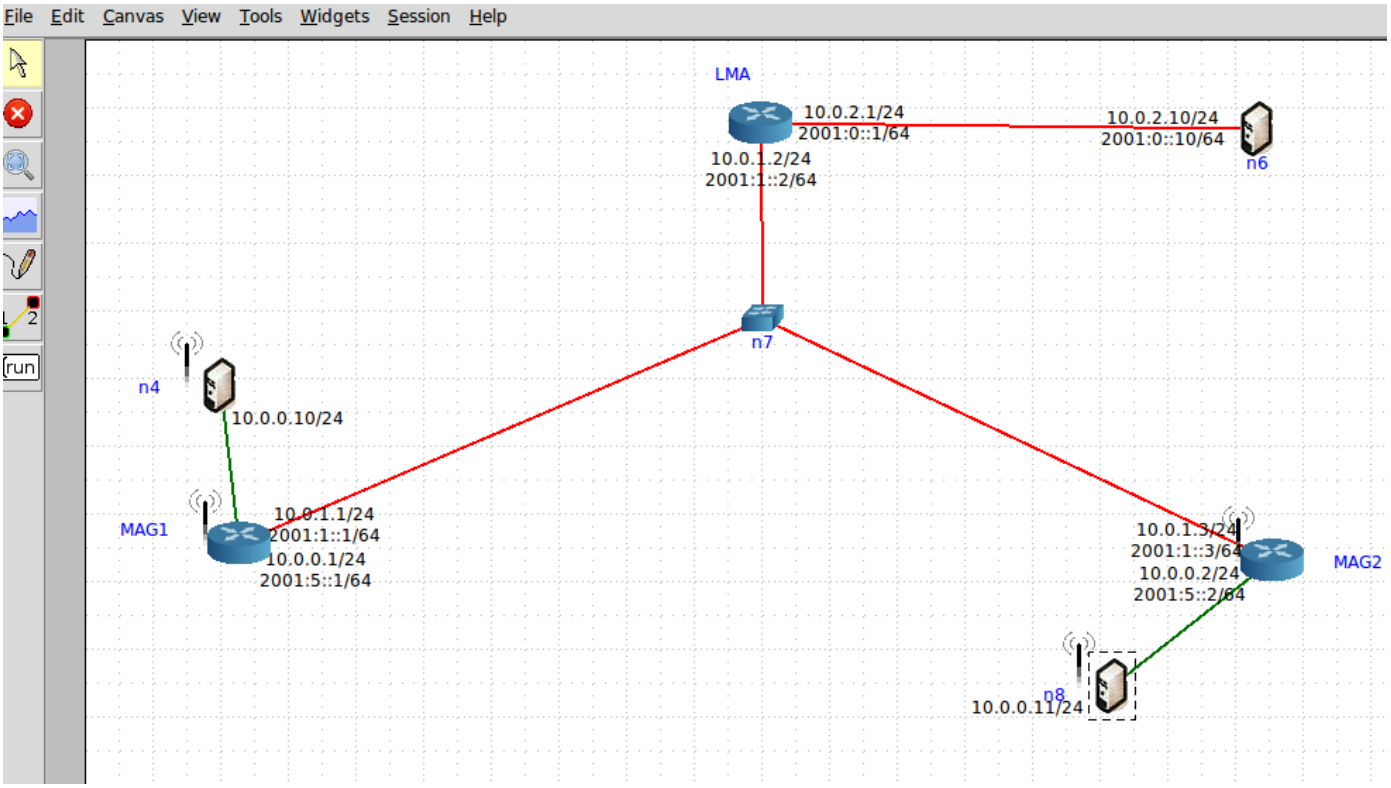


Figura 24: Escenario con MN conectados a distinto MAG

Esta prueba no resulto tan satisfactoria como las anteriores ya que como veremos en la siguiente imagen observamos perdidas de paquetes durante la ejecución del ping entre los MN. Esta pérdida sucede durante la actualización de la tablas de rutas del LMA, es decir, durante el intercambio de mensajes de señalización entre el LMA y los MAG's. Esto puede deberse a la capacidad de ejecución de la maquina anfitriona ya que, durante la actualización de la Binding Cache del LMA en ningún momento los nodos móviles quedan desconectados del dominio.

Los mensajes en blanco que vemos en la imagen corresponden al protocolo PMIP. La dirección IPv6 2001:1::1 es la correspondiente a la interfaz eth1 del MAG1 y la dirección 2001:1::2 es la correspondiente a la interfaz eth0 del LMA.

299	78.896298000	2001:5000::200:ff:fe00:35	2001:4000::200:ff:fe00:1	ICMPv6	158 Echo (ping) request id=0x0040, seq=77, hop limit=63
300	78.896320000	2001:5000::200:ff:fe00:35	2001:4000::200:ff:fe00:1	ICMPv6	158 Echo (ping) request id=0x0040, seq=77, hop limit=62
301	78.936703000	2001:4000::200:ff:fe00:1	2001:5000::200:ff:fe00:35	ICMPv6	158 Echo (ping) reply id=0x0040, seq=77, hop limit=63 (r
302	78.936726000	2001:4000::200:ff:fe00:1	2001:5000::200:ff:fe00:35	ICMPv6	158 Echo (ping) reply id=0x0040, seq=77, hop limit=62
333	104.493112000	2001:1::1	2001:1::2	NEMO	110 Binding Update
334	104.493629000	2001:1::2	2001:1::1	NEMO	110 Binding Acknowledgement
342	105.452872000	2001:1::3	2001:1::2	NEMO	110 Binding Update
343	105.453841000	2001:1::2	2001:1::3	NEMO	110 Binding Acknowledgement
348	106.106655000	2001:5000::200:ff:fe00:35	2001:4000::200:ff:fe00:1	ICMPv6	158 Echo (ping) request id=0x0040, seq=104, hop limit=63
349	106.106683000	2001:5000::200:ff:fe00:35	2001:4000::200:ff:fe00:1	ICMPv6	158 Echo (ping) request id=0x0040, seq=104, hop limit=62
350	106.147034000	2001:4000::200:ff:fe00:1	2001:5000::200:ff:fe00:35	ICMPv6	158 Echo (ping) reply id=0x0040, seq=104, hop limit=63 (
351	106.147062000	2001:4000::200:ff:fe00:1	2001:5000::200:ff:fe00:35	ICMPv6	158 Echo (ping) reply id=0x0040, seq=104, hop limit=62

Figura 25: Perdida de paquetes en Wireshark

Conexión de los MN en el mismo MAG.

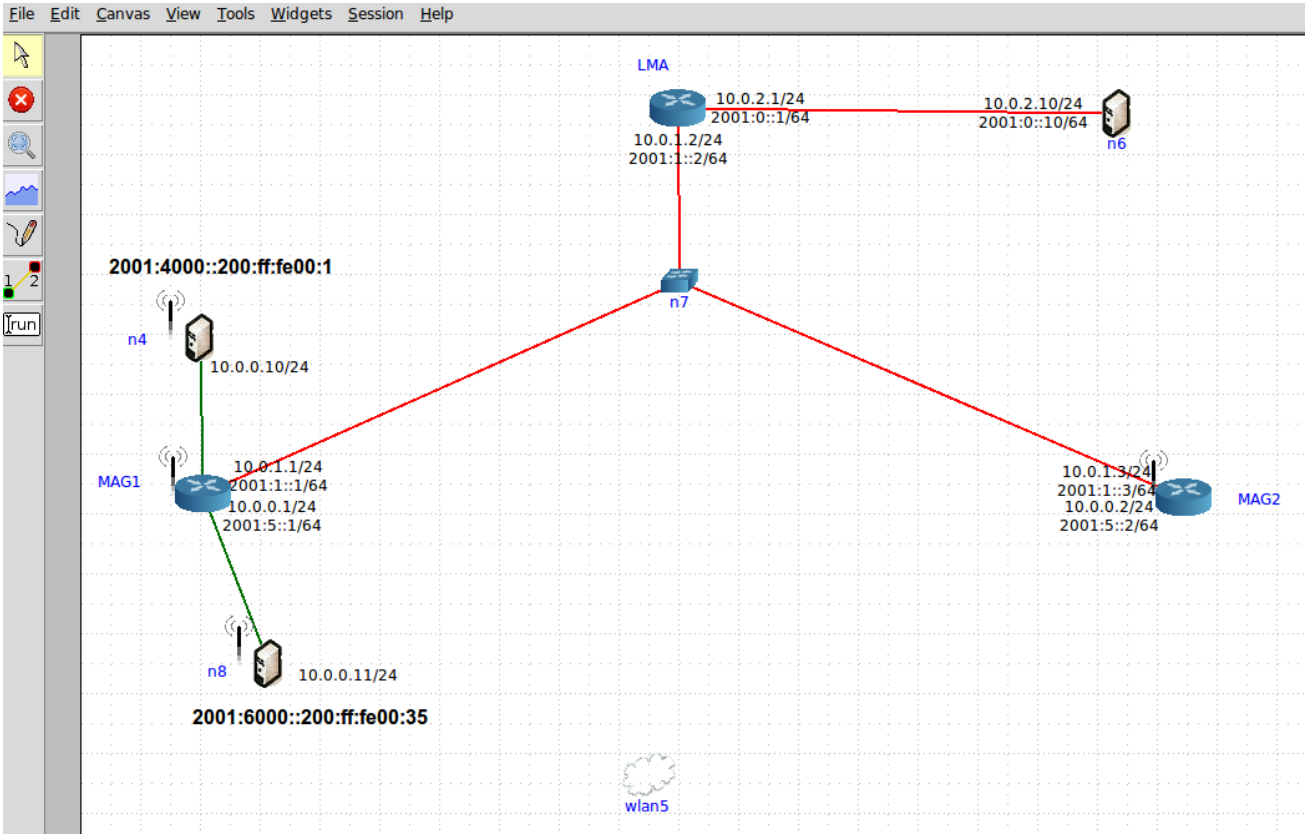


Figura 26: Escenario con MN conectados al mismo MAG

Esta prueba consiste en comprobar si los nodos móviles conectados al mismo dominio y al mismo MAG tienen conexión entre ellos.

Esta última prueba resultó fallida. Sin embargo pudimos identificar el problema y soplucionarlo, aunque debimos configurar los MAG's manualmente. El error se debe a que los MAG's configuran varias tablas de rutas.

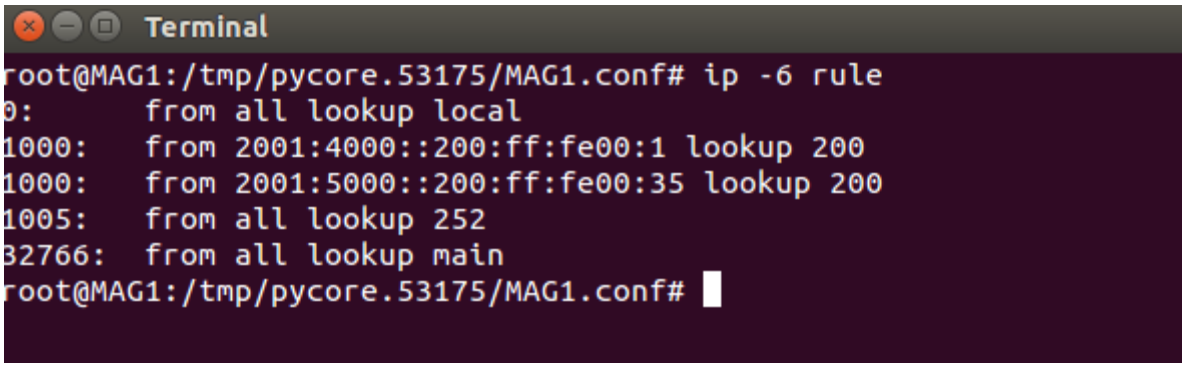


Figura 27: Tablas de rutas de MAG1

Como podemos ver en esta imagen el router que actúa de MAG configura varios tipos de tablas para enrutar su tráfico. Cuando un MN nuevo se conecta al MAG se crea una tabla de rutas nueva en la que se especifica como direccionar los paquetes con la dirección origen del MN.

Explicación de la imagen:

El numero a la izquierda de cada tabla es la prioridad con la que se van a leer las tablas. Este número va seguido de la especificación del origen de los paquetes. Una tabla solo se leerá si el paquete coincide con la dirección origen puesta en la tabla. La última palabra o número que se observa a la derecha es el nombre con el que nos referiremos a la tabla para modificarla o comprobarla. Como aclaración decir que la tabla que se modifica si no ponemos un nombre es la última, la llamada "main".

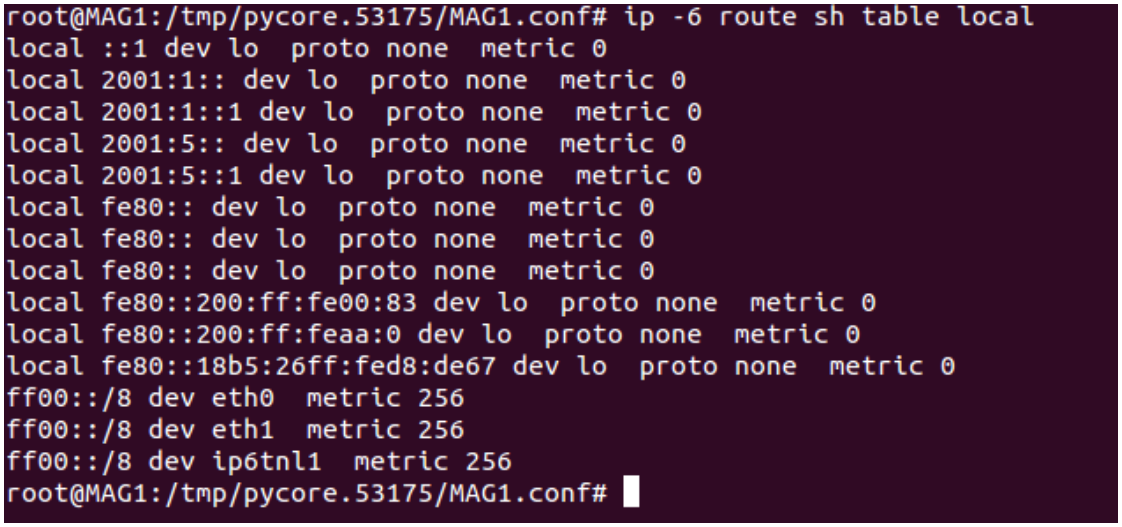


Figura 28: Tabla local de MAG1

Por lo tanto el MAG no leerá la tabla local a la hora de enrutar tráfico hacia los MN.

Así procedemos a comprobar la tabla 200 que es en la que se especifica cómo enrutar los paquetes con la dirección origen correspondiente a los MN.

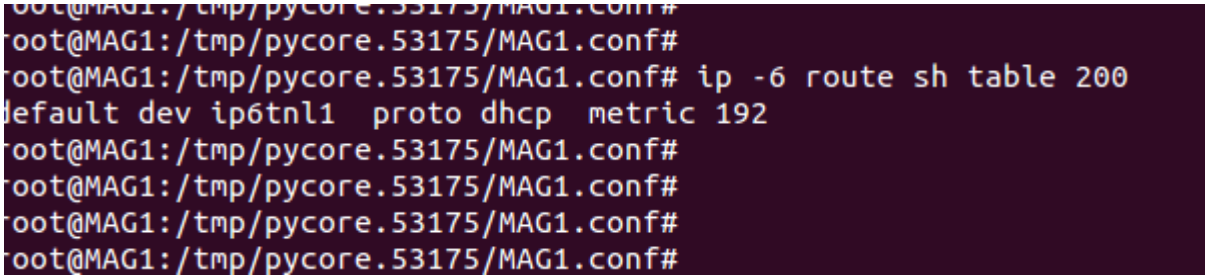


Figura 29: Tabla 200 del router MAG1

Esta tabla de rutas indica que los paquetes que vienen de cualquiera de los MN se envían directamente por el túnel al LMA. Esto implica que cuando un MN envía un paquete a otro MN conectado al mismo MAG este manda los paquetes al LMA en lugar de al otro MN.

La solución a este problema es introducir las rutas estáticas a los MN correspondientes en esta tabla de rutas 200. La forma de hacer esto es: “ip -6 route add 2001:4000::/64 dev eth0 table 200” y “ip -6 route add 2001:6000::/64 dev eth0 table 200” de tal forma que cuando el MAG recibe un paquete de cualquiera de los MN lo comprueba en esta tabla y lo envía directamente si el destino es uno de sus MN conectados.

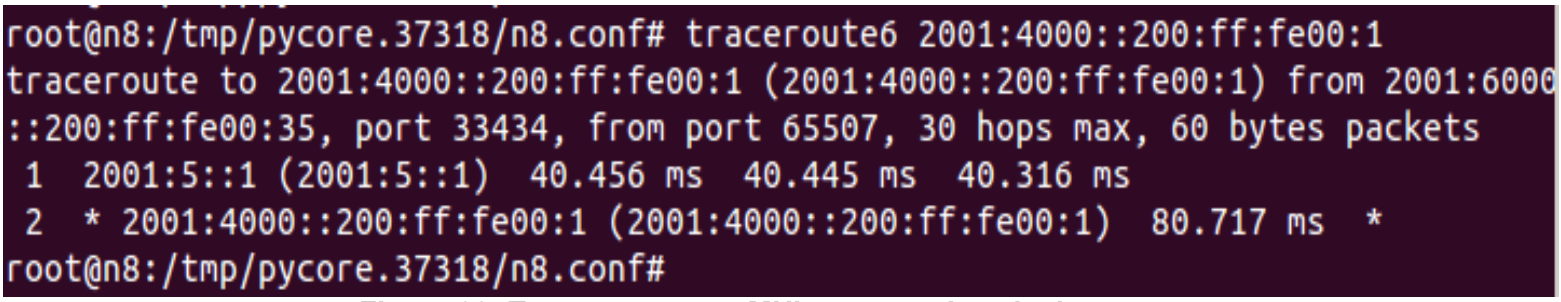


Figura 30: Traceroute entre MN's conectados al mismo MAG

En esta captura de pantalla podemos ver como hay conexión entre los dos nodos móviles que están conectados al mismo MAG. Sin embargo esta conexión no es estable ya que cuando realizamos un ping entre los dos nodos móviles se pierden bastantes paquetes. Este hecho no se sabe a que es debido ya que el MAG demuestra tener las tablas de rutas bien configuradas.

V.4 Conclusiones

Antes de pasar a las conclusiones del proyecto completo hay que decir que en general todas las pruebas realizadas se han comportado como cabía esperar eso sí, después de configurar y realizar los pasos de manera ordenada, ya que la realización de las configuraciones de los escenarios y el descubrimiento de los fallo no ha sido tarea fácil en muchos casos, dado que se debían a comportamientos diferentes a causa del emulador, como los envíos de RS desde las interfaces virtuales o los no envíos de RS desde los nodos móviles.

CAPÍTULO VI

CONCLUSIONES

VI. CONCLUSIONES

En este capítulo se explica el estado final del trabajo, los objetivos que no se han podido conseguir y si existe la forma de conseguir estos objetivos y como.

Las pruebas realizadas han resultado ser bastante satisfactorias dado que el código funciona razonablemente bien siguiendo la RFC. Como resultado de estas se han detectado desviaciones, es decir, no se sigue perfectamente el estándar, lo que causaría dificultades de interoperabilidad con otras implementaciones de PMIIPv6.

Estos fallos surgen de:

- **El código utilizado** se sale de lo establecido en la RFC; concretamente a la hora de la conexión de un MN nuevo, el MAG es el encargado de comprobar los permisos de acceso del MN al dominio, además propone al LMA un prefijo para asignar al MN y esto no debería suceder. El MAG únicamente debería enviar el identificador del MN al LMA y este comprobar los permisos de acceso y asignar un prefijo de dirección. Es un fallo menor dado que el resto de la ejecución, sí que mantiene lo establecido en la RFC.

- **A causa del emulador** hay que forzar a los MN a que envíen NA's a los MAG's. Esto debería suceder automáticamente como parte de la ejecución de IPv6. Esto es un problema de ejecución del emulador y no de PMIPv6 y es un inconveniente de poca importancia ya que con un simple comando, se consigue que el MN mande esta solicitud.

En definitiva podemos decir que se ha conseguido el objetivo del trabajo dado que:

- Se ha conseguido poner en funcionamiento, dentro del emulador de redes CORE, una serie de escenarios en los que se ha ejecutado el código de PMIPv6 con normalidad.

- Se han podido hacer pruebas de la ejecución y modo de funcionamiento del código de PMIPv6.

- Se han encontrado pequeñas diferencias entre el código utilizado y el estándar de la RFC de PMIPv6.

VI. CONCLUSIONS

In this chapter is explained the final state of the work, the objective shave not been achieved and if there is a way to achieve these objectives as explained.

Tests have proved quite satisfactory since the code works reasonably well following the RFC. As a result of these deviations have been detected, it is not perfectly follows the standard, causing difficulties interoperability with other implementations PMIIPv6.

These failures arise from:

- The code used is out of the established in the RFC; specifically when connecting a new MN, the MAG is responsible for checking the access permissions to the domain MN, also proposes to the LMA to assign a prefix to MN and this should not happen. The MAG should only send the identifier of the MN to the LMA and this check access permissions and assign an address prefix. It is a min or fault because the rest of the execution maintains the provisions of RFC.

- Because the emulator we have to force the MN to send NA's to MAG's. This should happen automatically as part of the implementation of IPv6. This is a problem of execution of the emulator and not of PMIPv6 and is a min or inconvenience since with a simple command, it is possible that the MN sends this request.

In short we can say that it has achieved the goal of work because:

- It has managed to operate, within the CORE network emulator, a number of scenarios in which your execute the code PMIPv6 normally.

- It has been possible to test the execution and operation mode code PMIPv6.

- Found small differences between the code used and the standard of RFC PMIPv6.

CAPÍTULO VII

LÍNEAS FUTURAS DE TRABAJO

VII. LÍNEAS FUTURAS DE TRABAJO

En este capítulo vamos a exponer algunas extensiones que se podrían llevar a cabo sobre este proyecto.

Como ya se señala en otro apartado de éste trabajo, CORE permite la creación de nuevos servicios [1]. Esto ya no es una labor sencilla de realizar, tampoco cambiaría en nada los resultados obtenidos, ni es imprescindible para el objetivo final del trabajo. Sin embargo sí que tendría grandes beneficios para usuarios menos experimentados ya que no habría que ejecutar el código de PMIPv6 desde un terminal. Esto implicaría que según el tipo de nodo se ejecutara el código con unas opciones u otras, es decir, dependiendo de si la ejecución es en un MAG o en un LMA, el servicio se ejecutaría con las opciones correspondientes.

Otra de las opciones para facilitar la ejecución de los nodos es configurar un servicio que ofrece CORE al que llama UserDefined, el cual está vacío y listo para rellenar por el usuario. De esta forma solo tendríamos que configurar el escenario la primera vez, dando una configuración diferente a este servicio en cada uno de los nodos, y guardarlo de tal forma que cada vez que lo abriéramos el servicio estuviese configurado y al ejecutar el escenario se iniciara automáticamente.

Una de las posibles mejoras a este trabajo consistiría en conseguir que los nodos móviles enviaran los RS automáticamente al conectarse a un punto de acceso de un dominio de PMIPv6, de tal forma que no hubiera que forzarlos manualmente a hacerlo y se pudieran conseguir pruebas de tiempo en el cambio de punto de acceso más fiables. Esto se podría hacer mediante un script que se ejecutara en los MN, al detectar cambios de enlace.

Otra de las pruebas que facilitaría mucho CORE seria la implementación de códigos diferentes de PMIPv6 en el mismo escenario. De esta forma se podría probar la interacción de las distintas implementaciones y ver si son compatibles.

Una de las posibilidades que ofrece este trabajo es poder probar otro tipo de protocolos de movilidad de redes IP tal forma que se pueda estudiar su ejecución y funcionamiento igual que se ha hecho con PMIPv6.

Por último se podría mejorar el código utilizado de tal forma que su ejecución se ajustara completamente al estándar.

Con todo ello llegamos a la conclusión de que el proyecto ha merecido la pena dadas las posibilidades de estudio que ofrece para el futuro.

CAPÍTULO VIII

PLANIFICACIÓN DE TAREAS Y PRESUPUESTO

VIII. PLANIFIACACIÓN DE TAREAS Y PRESUPUESTO

Este capítulo detalla los requisitos tanto de medios físicos (equipamiento) como de tiempo de trabajo dedicado.

VIII.1 Equipamiento

En nuestro caso, no ha sido necesario más que un ordenador portátil, con un Procesador Intel Core i5 cuyo coste puede ser de unos 550€.

VIII. 2 Tiempo de trabajo

El proyecto se realiza en cuatro fases:

VIII.2.1 Estudio del estado del arte: se analizaron los posibles escenarios de realización del proyecto: tipos de emuladores, escenarios físicos o virtuales y diferentes protocolos.

-Análisis del estado del arte.----------> 1 semana

VIII.2.2 Configuración del entorno de trabajo: Esto, aunque no es una de las partes principales del proyecto, sí es esencial dado que sin un entorno bien configurado, las pruebas no serian fiables e incluso no se podrían realizar. Esta tarea se realizó durante unas dos semanas.

Esta tarea se desglosa en cuatro sub tareas:

-Configuración y compilación del kernel de Linux. –---------> 10 días

-Instalación del código de PMIPv6. –----------> 2 días

-Pruebas en la maquina física. –----------> 4 días

-Instalación del emulador CORE. –----------> 2 días

VII.2.3 Pruebas del protocolo; Se realizan en cuatro fases:

-Generación del escenario ----------> 1 semana

-Primera ejecución ----------> 2 semanas

-Pruebas PMIPv6 ----------> 3 semanas

-Conclusiones ----------> 1 semana

VII.2.4 Realización de la memoria: Explicación escrita de todo lo realizado en el proyecto. Se definen también líneas futuras de trabajo y se detalla el presupuesto del proyecto.

-Realización de la memoria ----------> 2 semanas

- ✓ Total trabajo del ingeniero:

  Teniendo en cuenta que una semana equivale a siete días, tenemos un total de: 88 días.
  4 horas diarias durante 88 días.

  352 horas X 20 € = 7.040 €

- ✓ Total trabajo del Ingeniero sénior (Tutor encargado de la orientación en el trabajo):

  2 horas semanales durante 15 semanas.

  30 horas X 35 € = 1.050 €

VIII. 3 Costes indirectos

En esta sección se indican el presupuesto necesario para gastos como: conexión a internet, luz…

El coste estipulado para gastos indirectos es el 20% del coste del resto del proyecto. Por tanto tenemos unos gastos de:

8.640 * 0,2 ---------->1.728 €

Con los datos anteriores, el presupuesto económico más aproximado posible, para el desarrollo de éste proyecto sería de:

Equipamiento tecnológico……. 550 €

Remuneración labora total…….…….. 8.090 €

Costes indirectos……. 1728 €

TOTAL……..….. 10.368 €

CAPÍTULO IX

BIBLIOGRAFÍA

IX. BIBLIOGRAFÍA

[1] *Ignacio Soto, Carlos J. Bernardos, and María Calderó.* PMIPv6: A Network-Based Localized Mobility Management Solution. The Internet Protocol Journal, Volume 13, No.3. September 2010.

[2] Ley General de las Telecomunicaciones (LGTel), Jefatura del Estado. BOE-A-2014-4950, 10 de mayo de 2014. **https://www.boe.es/buscar/act.php?id=BOE-A-2014-4950**

[3] **https://www.itu.int/osg/spu/imt-2000/technology.html#Cellular Standards for the Third Generation** , visitada por última vez en septiembre de 2016

[4] CORE Documentation. Obtenida el 5/junio/2015, de **http://downloads.pf.itd.nrl.navy.mil/docs/core/core_manual.pdf**

[5] **http://www.ubuntu-es.org/** Visitada por última vez en septiembre de 2016

[6] **http://www.freebsd.org/** Visitada por última vez en septiembre de 2016

[7] A.C.C. Capítulo VI capa de red. (n.d.) **http://www.ie.itcr.ac.cr/acotoc/CISCO/R&S%20CCNA1/R&S_CCNA1_ITN_Chapter6_Capa%20de%20red.pdf**

[8] **http://www.angelfire.com/wi/ociosonet/29.html** Visitada por última vez en septiembre de 2016.

[9] S. Gundavelli, Ed. K. Leung, V. Devarapalli, K. Chowdhury, B. Patil. Proxy Mobile IPv6. RFC 5213 (Proposed Standard), August 2008.

[10] R. Hinden, S. Deering. IPv6, RFC 3513 (Proposed Standard). April 2003

[11] S. Frankel, S. Krishnan. IPsec, RFC 6071 (Informational). February 2011

[12] **http://openairinterface.eurecom.fr/openairinterface-proxy-mobile-ipv6-oai-pmipv6** Visitada por última vez en septiembre de 2016.

[13] **https://www.wireshark.org/**. Visitada por última vez en septiembre de 2016.

ANEXO I

INTRODUCCIÓN

ANEXO I

En este punto, vamos a explicar con detalle cómo se ha hecho la configuración de la máquina física en este trabajo.

A) Configuración del Kernel de Linux.

- Descargar el código fuente.
- Configurar el kernel.
- Compilar e instalar el kernel y sus módulos.
- Actualizar el grub.

Para el correcto funcionamiento del emulador, no es necesario hacer una configuración especial pero, para el funcionamiento del código de PMIPv6, hay que activar algunas de las opciones que por defecto están desactivadas. Estas opciones son las siguientes:

CONFIG_EXPERIMENTAL=y

CONFIG_SYSVIPC=y

CONFIG_PROC_FS=y

CONFIG_NET=y

CONFIG_INET=y

CONFIG_IPV6=y

CONFIG_IPV6_MIP6=y

CONFIG_XFRM=y

CONFIG_XFRM_USER=y

CONFIG_XFRM_SUB_POLICY=y

CONFIG_INET6_XFRM_MODE_ROUTEOPTIMIZATION=y

CONFIG_IPV6_TUNNEL=y

CONFIG_IPV6_ADVANCED_ROUTER=y

CONFIG_IPV6_MULTIPLE_TABLES=y

CONFIG_IPV6_SUBTREES=y

CONFIG_ARPD=y

CONFIG_INET6_ESP=y

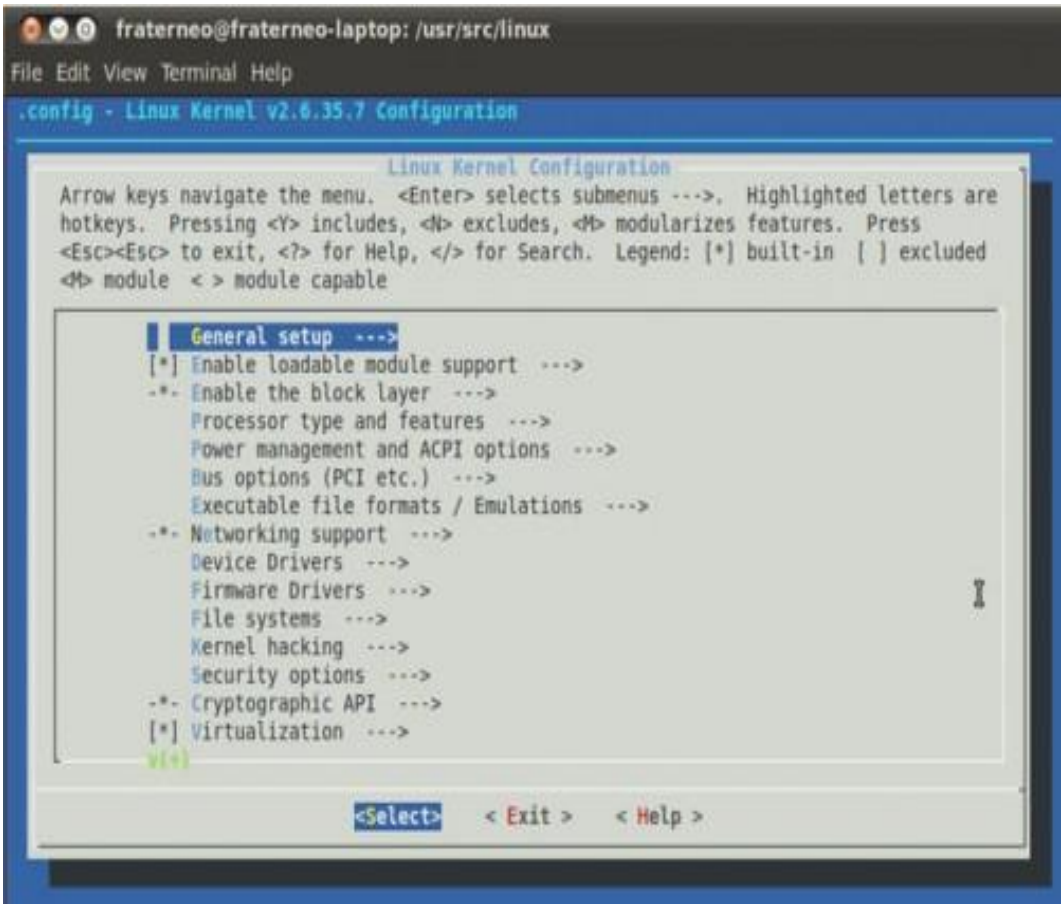
CONFIG_NET_KEY=y

CONFIG_NET_KEY_MIGRATE=y

- Lo primero de todo es descargar el código fuente de la versión del kernel de Linux 4.1.1. Como Linux es software libre, se puede descargar de muchos sitios pero lo ideal es bajarlo de los repositorios oficiales de Linux. **“https://www.kernel.org/”**

- A continuación, debemos configurar el kernel; Esto lo haremos con el comando “make menuconfig”.

  Esta es la forma más visual de configurar el kernel. Nos aparecerá una pantalla así:



Terminal para configurar las opciones del kernel.

La forma más fácil de encontrar las opciones es buscándolas, para ello presionamos </>. En ocasiones las opciones no se llaman exactamente como se especifica arriba pero el buscador explica donde se pueden encontrar.

- Una vez configuradas todas las opciones, debemos proceder a compilar e instalar el kernel y sus módulos. Esto se ejecuta con los siguientes comandos:

  "make"
  "make modules"
  "make modules_install"
  "make install"

Por último hay que actualizar el grub:

"update-grub"

De esta forma podremos elegir nuestro nuevo kernel cuando arranquemos Linux.

En nuestro caso, tenemos instalados dos sistemas operativos en la maquina, por lo que al encenderla, debemos elegir el que vamos a iniciar. Al elegir la partición del disco duro en la que está instalado el SO de Linux, tendremos varias opciones; entrando en opciones avanzadas, nos aparecerán los diferentes kernel instalados y ahí elegiremos el 4.1.1.

Una vez iniciado, debemos comprobar si nuestra configuración es válida para el correcto funcionamiento de PMIPv6. Esto se realiza con el comando: "./chkconf_kernel.sh".

B) Instalación y ejecución de PMIPv6

Una vez configurado correctamente pasamos a la instalación y **ejecución de PMIPv6.**

Estando en el directorio principal del paquete de PMIPv6, ejecutaremos el comando "make" el cual compilará el código; Una vez compilado, solo nos queda ejecutarlo en el emulador.

Para ejecutar PMIPv6, debemos tener en cuenta, si lo vamos a hacer en un MAG o en un LMA, ya que el comando cambia.

- Ejecución en un MAG:

Estos tres comandos habilitan el encaminamiento en el MAG.

```
sysctl -w net.ipv6.conf.eth1.forwarding=1
sysctl -w net.ipv6.conf.eth0.forwarding=1
sysctl -w net.ipv6.conf.all.forwarding=1
```

Cargamos el módulo de túneles IPv6.

```
modprobe ip6_tunnel
```

Ejecutamos el código con las opciones:

-**m** Implica que el nodo se comportará como un MAG.

-**L** Indica la dirección IPv6 de la interfaz del LMA a la que se conecta el MAG.

-**N** Indica la dirección de la interfaz del MAG que se conecta con el LMA.

-**E** Indica la dirección de la interfaz del MAG a la que se van a conectar los MN.

-i Indica que se crearan túneles de conexión entre las dos interfaces anteriores.

-**d<número>** Indica el nivel de análisis que se va a exponer en la pantalla durante la ejecución.

/usr/local/sbin/pmip6d -m -L 2001:0::1 -N 2001:1::1 -E 2001:0::2 -i -d10

- Ejecución en un LMA

Estos tres comandos habilitan el encaminamiento en el MAG.
sysctl -w net.ipv6.conf.eth1.forwarding=1
sysctl -w net.ipv6.conf.eth0.forwarding=1
sysctl -w net.ipv6.conf.all.forwarding=1

- Cargamos el modulo de tunelesIPv6.

modprobe ip6_tunnel

Ejecutamos el código con las opciones; En este caso solo necesitamos indicar que la ejecución se realiza sobre un LMA "-a" y la dirección de la interfaz de LMA a la que se conecta el MAG. El resto de opciones son las indicadas anteriormente.

/usr/local/sbin/pmip6d -a -L 2001:0::1 -i -d10

El comando utilizado para el envío de NA desde el MN al MAG es:

rdisc6 -1 -r2 eth0

-1 Implica que en el momento que se recibe una respuesta se para la ejecución.

-r2 Indica que se van a realizar dos envíos de este mensaje antes de finalizar. eth0 indica la interfaz por la que se van a mandar los mensajes.

ANEXO II

ANEXO II. PALABRAS CLAVE

Vocabulario técnico utilizado en el trabajo.

Sistema operativo (SO): Programa inicial que ejecuta una máquina, es decir, programa sobre el cual se van a ejecutar el resto de programas y que controla los procesos básicos.

Linux: S.O. de en software libre.

FreeBSD: S.O. de en software libre.

Protocolo: Sistema de reglas que permiten que dos entidades de un sistema de comunicación, se comuniquen entre ellas para transmitir información.

-IP - PMIPv6 - MIPv6 - OSPF - DHCP - SSH

RFC (Request for Commentes): Publicaciones en internet en las que se explican los protocolos**.**

MAC: Direcciones físicas asociadas a los dispositivos.

KERNEL: Es el componente central de un sistema operativo; Se encarga de manejar los recursos hardware como la CPU, la memoria y los discos duros, y proporciona abstracciones que le dan a las aplicaciones una visión consistente de esos recursos.

GUI: Son las siglas de "Graphical User Interface". Es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz.

Máquina anfitriona: Se denomina así, a la maquina física que alberga una maquina virtual.

Máquina virtual: Es un programa que se ejecuta en un ordenador y aparenta ser una maquina totalmente independiente de la maquina anfitriona.

Python: Es un lenguaje de programación interpretado, cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional.

El CORE (common open research emulator): Es un emulador de redes, basado en código Linux.

DefaultRoute: Servicio de CORE que asigna al router por defecto a los dispositivos.

IPForward: Servicio de CORE que posibilita el encaminamiento

Dominio: Es la red gestionada por el LMA

LMA (Local Mobility Anchor): Nodo que da acceso, al resto de la red, a todos los dispositivos conectados a su dominio.

M**AG (mobile Access Gateway):** Nodo que da acceso al dominio, de los dispositivos.

MN (mobile node): Dispositivos inalámbricos conectados a la red.

LMAA (LMA Adress): Dirección IP dada a la interfaz del LMA que se une al MAG.

Proxi-CoA (Proxy Care-of Adress): Es la dirección asociada a la interfaz del MAG que lo une con el LMA.

PBU y PBA: Mensajes que pertenecen al protocolo de PMIPv6.

Router: También conocido como enrutador o encaminador de paquetes— es un dispositivo **que** proporciona conectividad a nivel de red o nivel tres en el modelo OSI.

Switch: Es el dispositivo digital lógico de interconexión de equipos que opera en la capa de enlace de datos del modelo OSI.

SSID: Es el nombre que identifica una red inalámbrica.

Binding cache: Es la tabla de rutas del LMA en la que almacena una entrada para cada nodo móvil conectado a su dominio.
Prefijo IPv6: Primera parte de una dirección IPv6 que especifica de que tipo es y a que red pertenece. Este prefijo cambiara de tamaño según la máscara de red asignada.

Script: Es un archivo de órdenes o pequeño programa con el que poder automatizar ciertas tareas en un ordenador.

OSI: Es un modelo de referencia para los protocolos de la red de arquitectura en **capas**.

Capa OSI: Cada uno de los niveles del modelo de arquitectura OSI.

Ping: Es un comando o una herramienta de diagnóstico que permite hacer una verificación del estado de una determinada conexión de un host local con al menos un equipo remoto contemplado en una red de tipo TCP/IP.

Flags: Uno o más bits que se utilizan para almacenar un valor binario o código que tiene asignado un significado.

Hub: Es el dispositivo que permite centralizar el cableado de una red informática.

Host: Máquinas conectadas a una red, que proveen y utilizan servicios de ella.

ANEXO III

ANEXO III. SUMMARY IN ENGLISH

In recent decades the use of mobile devices has increased very significantly, so much so that nowadays most people fewer than 40 makes using mobile devices essential for their lives.

This is due to the advancement of technology emerged in recent years has allowed us to carry small and lightweight computers in the pocket, so that from anywhere and at anytime we can send a message to someone else or write what we are doing, even documenting it with photos and videos.

As the technology has advanced, users in turn have become more demanding with its use, that is, at the same time they demand more services and want them to work better every day. From here it arises one of the biggest challenges for these new technologies. Getting a consistent and uncut mobile communications.

During the time when the phones used first and second generation nets like the GSM network, that was just to make voice calls and text messages (SMS), this was not a problem as the connection with the access points was simple and, even if there was no connection at that time to send a message, it was simply sent later. But the problem comes when a device needs to be constantly connected to the network due to the use of their applications and, in addition, such device is constantly in motion changing the access point, as the change of access point of data networks based on the IP protocol, such as the LTE networks, is not something trivial.

This is where it appears the need for protocols such as MIPv6, NEMO or PMIPv6, which are able to manage the mobility of devices connected to third generation networks, denominated 3G networks, or subsequent ones.

The management of the mobility of these devices is a must since, apart from the possible use that can be given to these devices as web browsers, which can use other technologies for this service, the Internet network which is connected to our device needs to know the location of it to send the messages than other devices send it.

Because of this, it emerges the need to prove relatively new protocols as PMIPv6, in which mobility management is done by the network itself, thus our device does not intervene in it. Like this, we get to be reachable by the network without installing additional software on mobile devices.

At this point is where it comes into scene our project, as its main aim is to check that the protocol PMPv6 can run in a networks' emulator, being its functioning absolutely equal to the functioning of a real network.

Prior to explain the developing made and the results obtained it is necessary to explain the analysis of possible tools assessed before starting with this piece of work.

The final objective is to probe the protocol, but these tests could have been realized in scenarios with several physical machines rather than in one single machine with one emulator installed.

The decision of running the protocol in one single physical machine came determinate mainly by the ease of use and the possibilities than that offered. A physical scenario makes more expensive the tests since to run the protocol in the most basic scenario it would be necessary four computers, one switch and cables to link the computers with the switch. Furthermore, it exists the possibility of failure due to the bad connection of some of the cables. On the other hand, the use of the physical machines implies a limitation on the possibilities of creation of scenarios, as normally we do not have a physical space to work. These two problems were solved by the use of one emulator which is run in a one single physical machine.

Moreover, it exists the possible use of a simulator rather than an emulator. It was concluded that the best option was to use an emulator as the run mode does not depend on the host machine and the results obtained would be much more close to reality.

Finally, it was taken the decision of using an emulator CORE due to its ease of use, since the project has as objective, although secondarily, its use for education. To make it easier, it was found appropriate an easy to use environment. On the other hand, the choice of CORE came determined by the fact that it can be installed and used in one machine with an operating system LINUX and, at the same time, the used code for the implementation of PMIPv6 is made to work on LINUX as well. Other fact that leaded us to the choice of CORE is its mode of working with virtual small machines. CORE creates a virtual light machine based on LINUX, what allows us to run programs in the nodes of the scenarios created in the emulator and that these programs are independent from one another. Like this, it will be possible to run the PMIPv6' code in every of the nodes independently.

After all of this, it was preceded to the study of the socioeconomic frame, as well as about the wireless networks' regulator frame in Spain and Europe; more concretely, about the third generation wireless networks, like the LTEs.

Regarding to the socioeconomic frame, as was already said, the technology has increased at a great speed in the latest years, what leaded us to the nearly imperative use of the mobile phone. The biggest change affecting the telephone companies, to which they have had to get adapted, has been the replace of the old services by the new ones, as these are more complete and cheaper for the vast majority of the regular users, not happening the same in the business ambit.

Hence, the broadband mobile communications sector, such as the 3G networks, has been object of regulation by the organizations, not only European, but worldwide as well. The fact that the new technologies have had a so high level of penetration into the market implies the possibility of monopolies by the companies that have their own communications network. That is why the regulation bodies of each country obligates these companies to rent their networks to other companies; and to do so in a reasonable cost, so that the final users can choose among the different operator without differences relating to the quality of the service offered, as it has to be the same in any geographic point.

Once seen the technologies used and the legal framework, we move to explain the parts, tests and results of the project.

The first thing to be done, once decided where the tests were going to be carried out, was the configuration of the environment. Due to the emulator was going to be run in one single machine, and that in this was going to be run both mobile devices and network' routers, we had to prepare the machine for the PMIPv6 run. For that, we needed to activate some options of LINUX's kernel, apart from one specific version of kernel 4.1.1. To activate these options, it was necessary to compile the kernel and to install it in our host machine. Once done this and checked that the options were activated, it was preceded to the emulator CORE installation, the PMIPv6 installation and the Wireshark installation, which is a very useful programme to check the network traffic.

It has to be highlighted that although the CORE and Wireshark installations are something trivial, as they are both more used programs and getting installation files is not difficult, the PMIPv6' code installation was a little more tedious since we had the font code and it was necessary to export it to look for the way to install it and, later on, check how it was run. However, the most complicated and what more time consumed was the compilation of the LINUX's kernel: first it was necessary to search for download the kernel in its version specified above, and then, during the compilation, it had to find al the options that had to activate. Although said like this it can seem easy, it was not since the options offered by LINUX when compiling are many whereas the options needed were very specific.

Once configured the machine, it was preceded to its testing. It was carried out through a little run of the code on it, checking like this than the code was run without errors. It was

done to avoid confusions in case that, when testing the emulator, it gave some type of failure, discarding so a configuration failure.

When the machine was configured and tested, it was created the first simple testing scenario. In this scenario, there were two routers that worked as access routers; one router managed the PMIPv6' domain; and two host ones, one connected to the PMIPv6' domain and other connected to the routed in charge of managing the network.

The mobility simulation is done through the use of wireless networks that CORE offers, so that connecting the access routers to this network, anyone can provide service to host or other routers connected to the same wireless network.

The access routers had to be configured with one MAC address in the wireless interface equal in both, what is one of the specifications of the RFC 5213. In addition, it had to be deleted the IPv6 address assigned to the mobile device as this address must to be configured by the protocol. The last thing done before running the protocol was the creation of a file called *match*, in which the prefixes that are going to be assigned to the mobile nodes, and the MAC addresses of the mobile nodes that are going to have access permitted to the domain, will be placed. Thus, it had to be assigned a MAC address, equal to the one written in the file *match,* to the mobile node.

With the configuration of the scenario finished, it was passed to the code run. The first thing observed was than the access routers or MAGs received constantly messages from confusing origin addresses, as they did not match with any of the nodes created in the scenario. Later it was discovered than these messages were sent from the virtual interfaces belonging to the interfaces created by the host machine to every of the interfaces created by the emulator.

The second test made on the same scenario consisted in checking if the mobile nodes could acquire a IPv6 address when connecting with the accesses points of the PMIPv6' domain. It turned out that to these to acquire an address they have to send a Router Solicitation message, message whose delivery had to be automatic; however, due to the emulator run, the delivery of these messages must be done manually, so that while the message to the mobile node is not sent, the IPv6 cannot be acquired.

Once the node belonged to the PMIPv6' domain, it was tested if it had connection with the node connected to the router that managed the domain, the LMA, and could be tested that the packets sent by the ping mechanism got to destination without any kind of problem.

It could be considered that the tests are the initials to be able to affirm that the code runs satisfactorily in the emulator, but it had to go on making tests so that it was tested the protocol followed the standardized by the PMIPv6' RFC.

The tests to check if the code used followed the protocol were the followings:

- Testing of the messages interchange between the LMA and the MAG; and between the MAG and the mobile node, during the connection of one new MN to the PMIPv6' domain.
- Testing of the messages interchange during the disconnection of one MN between the MAG and the LMA.
- Analysis of the messages when one MN changes the access point inside the same domain.
- Connection between MNs connected to the same domain in different MAGs.
- Connection between MNs connected to the same MAG.

All these tests were made using the tool Wireshark, which lets see the network traffic of the chosen interface. As, in our case, all the interfaces were inside the same machine, Wireshark allowed us to see all the messages interchange of our emulated network.

The majority of the results obtained were positive despite some complications emerged due to the code implementation or the way of running the emulator. The only differences appreciated between the code used and the standard one of RFC 5213 were in the functionality of the MAGs, as these check the access permissions to the network of the MNs and send the prefixes that must to be assigned to the mobile nodes. The first task is made by the LMA and the second one should not happen as the MAG must only send the identifier to the LMA and this one assigns the prefixes.